
MAGAZINE
MSX
AÑO I
Núm. 4
Agosto 1985
250 Ptas.
TOSHIBA
mass storage
EERON
BYTE
operating system
BOOT
BIT
NIBBLE
LA JERGA INFORMATICA
Comunicación entre ordenadores
JOYSTICKS, ¿CUALES SON?

¡EL IMPERIO CONTRAATACA!

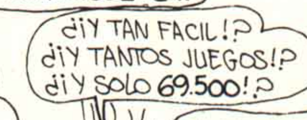
¡¡BANZAÏ SAMURAÏ!!











Ordenador Personal

TOSHIBA HX-10

Su Ordenado Servidor

69.500 Ptas.





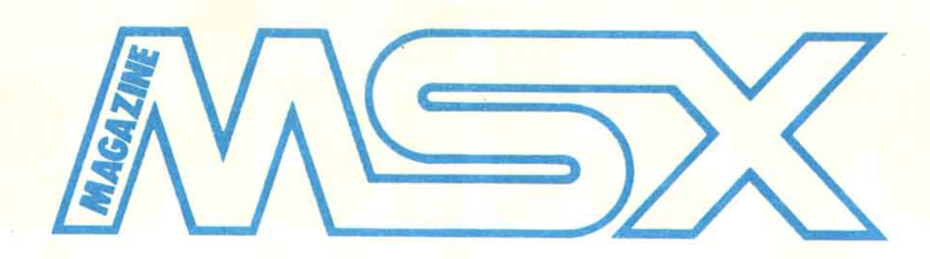
Características principales:
Sistema standard MSX. Memoria de 64 K RAM, 32 K ROM y 16 K de pantalla. 16 colores. 73 teclas. 32 sprites. Sistema multicolor: 64 x 48 bloques. Sonido: 8 octavas tres acordes. Conexiones para: casette, impresora, 2 mandos y futuras expansiones.



TOSHIBA
española de microordenadores s.a.
Caballero, 79 · Tel. 321 02 12 · Telex 97087 EMOS · 08014 BARCELONA



El sistema M... un standard utilizado universalmente que permite disponer de una gran variedad de programas y accesorios compatibles entre sí



AÑO I - Núm. 4
Agosto 1985

El mes de agosto se presenta caluroso y tranquilo, no se aprecia movimiento en el mercado y los fabricantes ya están pensando en la estrategia a seguir a la vuelta del veraneo.

Buen momento para aprovechar las horas muertas frente al ordenador, hemos dedicado este número a programas para el pequeño de la casa. No significa que los mayores caigan en el olvido, al contrario, dentro de este número encontrará, en el artículo de Terminología Informática, una curiosa forma de quedarse con los amigos o de entender las explicaciones del que nos quiera vender un ordenador a buen precio.

¡Buen veraneo y hasta septiembre!

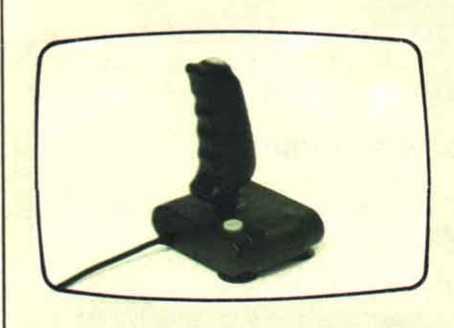




Joysticks, los amigos

NAVAZO

Quickshot I.



Fuerte y duradero. Con estas dos palabras podríamos resumir, sus características principales. Posee dos botones de disparo, uno en la base y otro al final de la empuñadura, siendo esta anatómica.

Sólo pudimos achacarle un pero. Después del uso y abuso al que ha sido sometido, el disparador superior dejó de funcionar correctamente, esto lo consideramos totalmente normal, ya que ha sido el peor tratado de todos los que han pasado por esta redacción.

Su agarre a la mesa, por cuatro ventosas, permite tirar de un lado hacia el otro sin problema alguno. Aconsejable para todos aquellos que son partidarios de tener uno para la gente menuda de la casa.

Es preciso en los juegos de acción y su maniobrabilidad es bastante buena.

Hace algún tiempo ya, que los joysticks han pasado de ser meras palanquitas, accesorios sin importancia, a convertirse en el brazo ejecutor de aquellos usuarios a quienes el teclado no convence cuando se trata de juegos de acción.

Actualmente son los periféricos más populares del mercado, creciendo la variedad y las versiones, algunos con mejores prestaciones que otros.

incansables

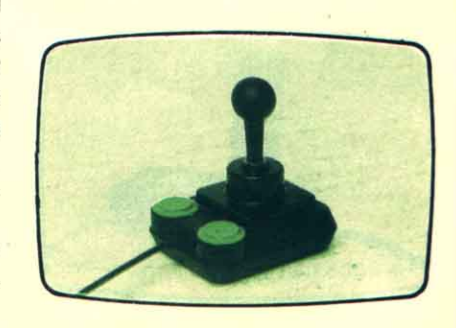


Los MSX, aparte de tan hablado y comentado tema de la compatibilidad, tiene otro gran punto a su favor, cuenta con cuatro teclas que permiten controlar el cursor. Estas, junto con la barra espaciadora, se han convertido, gracias a los juegos, bélicos o no, en compañeros inseparables.

Pero también tienen otra función, facilitar la operación de editar por pantalla. Esta característica permite corregir los programas directamente en la pantalla, sin tener que editar línea a línea. Por el momento, este punto no nos interesa, ya que cualquier usuario que se compre un *joystick* desde luego no será para editar y corregir programas.

El problema se plantea a la hora de elegir el tipo idóneo a nuestras necesidades. La mayoría de los juegos del mercado se pueden utilizar con *joystick*, pero ¿cuál es el mejor? ¿qué tipo de *joystick* es el más adecuado? No vamos a decir que tal marca es mejor que tal otra, ni que la capacidad de disparo de una es sen-

Kempston.



¿Tiene algún martillo a mano? Lo va a necesitar, puesto que es duro como una roca. En los juegos de acción se utilizó hasta el límite. Este gustó especialmente porque aguantó golpes y tirones que otro no habría aguantado.

La empuñadura, en forma de bola, es distinta a lo que se suele ver, sin embargo, una vez acostumbrado a la forma, no existe problema alguno. Posee dos disparadores en la base, característica que agradecerán los jugadores zurdos.

Lástima que su apoyo en la mesa sea tan pobre, puesto que al no tener ventosas, jugar con él se convierte en ejercicio de equilibrio.

De cualquier manera, al tener los dos disparadores en la base del mando, tendrá que utilizar las dos manos, una que controlará el *stick* y la otra para sujetar la base a la vez que dispara. Aunque parezca una operación compleja, no lo es tanto. Repetimos, es cuestión de acostumbrarse.

Su maniobrabilidad es precisa y muy buena. Ideal para tener el joystick eterno.

Quickshot II.

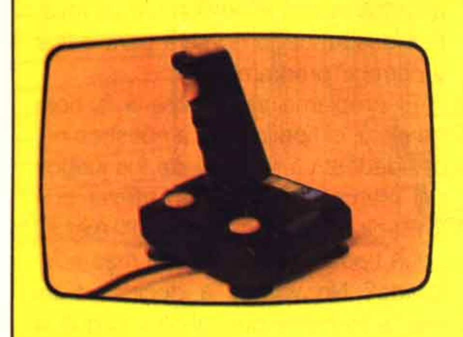


Si la primera versión de este *joystick* es buena, esta es la superación del anterior. Más completo y complejo, parte del modelo anterior (Quickshot I), con dos botones de disparo dispuestos en la parte superior del mando, uno en forma de gatillo y otro encima

El mando es anatómico y se adapta muy bien a la mano, siendo cómodo y fácil de manejar. La plataforma sobre la que viene el *joystick*, es sólida y desde luego, no parece que se vaya a deteriorar si se abusa mucho. Además del soporte típico de las ventosas, tiene otra particularidad que le caracteriza. En la parte trasera de la plataforma, hay un *switch*, que permite alterar la velocidad de disparo, una posición para el tiro a tiro y otra de tiro permanente. Al igual que en la versión anterior, se deterioró un disparador cuando se utilizó al límite, concretamente, el que está situado en el gatillo.

De cualquier manera, es preciso y la maniobrabilidad es bastante buena.

Sakati.



De tacto similar al Quickshot II y con una buena presentación, este joystick se caracteriza por los cuatro pulsadores que posee y por la solidez que tiene. El mando es anatómico con dos pulsadores, uno en forma de gatillo y otro en la parte superior, pero además cuenta con dos disparadores más en la base de la palanca, lo que significa que, para disparar no tendrá problemas.

Es preciso, de respuesta rápida y tiene buena maniobrabilidad. En el juego de acción probado, se le exigió hasta el límite y respondió a la perfección. Con cuatro ventosas en la base y siendo esta ancha, el *joystick* está más que asegurado. Moverlo no requiere demasiado esfuerzo y aunque parezca que se va a romper, nada más lejos de la realidad, su buena constitución aguantará todo lo habido y por haber.

Los cuatro disparadores, son de suma utilidad, puesto que no siempre vamos a utilizar los que se encuentran sobre la empuñadura, y permiten las más diversas posibilidades.

Canon VJ-200.



Es un joystick estilizado, de líneas rectas y de buen tacto. Es preciso y cómodo, con dos disparadores, uno en la base y otro en la parte superior de la palanca, da la sensación de romperse, no es motivo de asustarse, ya que al momento, uno se acostumbra y a la larga lo agradece.

La base se apoya sólidamente en cuatro ventosas que se agarran a cualquier tipo de superficie, permitiendo innumerables movimientos bruscos, sin temer por su integridad.

En los juegos de acción se comportó correctamente, sorprendiéndonos por su precisión y la rapidez de respuesta a nuestros requisitos. Su maniobrabilidad es uno de sus puntos fuertes.

La empuñadura, aunque no sea anatómica, es fácil de sujetar y el disparador situado en la parte superior es sencillo de accionar.

Sony JS-55.

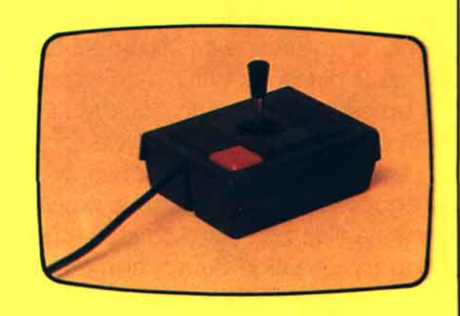


El diseño, lo más atractivo de este aparato, es vanguardista, lo que obligará a los usuarios potenciales a probar una y otra vez, hasta cogerle el tacto.

De los utilizados, es el más sensible, teniendo que poner especial atención a una serie de detalles, como son la plataforma, debido a la nueva posición del mando.

Tiene tres disparadores, dos a los lados del mando y uno en la base. En la parte posterior de esta, hay un interruptor que permite elegir entre uno de los dos disparadores existentes en el mando del *joystick* que le permiten al jugador zurdo jugar comodamente. Dio la sensación de no ser muy sólido y no era muy preciso en los juegos de acción. Esto último, lo achacamos al diseño ya que al no estar acostumbrados a la forma, era difícil controlar el objeto. La base también planteó algún problema, al ser tan estrecha hacia el juego más de equilibrio que de otra cosa. De cualquier manera, está muy logrado y el diseño no debe retraer al comprador. El uso frecuente es lo que permitirá mejorar las acciones que en el juego se tengan que realizar.

Philips, VU 0001.



En su forma y concepción, recuerda mucho a los primeros que aparecieron en el mercado junto con aquellos video juegos de antaño.

Es muy simple, con una base sin sujeción y un solo disparador.

Util para todos aquellos que no pretendan exigir el máximo esfuerzo del aparato.

El mando para dirigir el objeto consta de una pequeña palanca situada en la parte superior de una semi-esfera, mientras que el disparador queda algo alejado de los dedos, lo que complica su manejo con una sola mano. De cualquier manera, al no tener ni gomas en la base, ni ventosas, tendrá que usar las dos manos. Al contrario que modelos anteriores, donde podíamos o no usar las dos manos, aquí si va a hacer falta, puesto que si nos limitamos a usar solamente una, se nos complicará demasiado el juego, por muy simple que este sea.

En los juegos de acción, dejó mucho que desear. Su principal problema radica en que el movimiento queda restringido a ocho direcciones y la base no tiene sujección.

Challenger.



Su color plateado, no le hacen pasar desapercibido. Al contrario, se hace notar, por su forma y maniobrabilidad. Esta permite hacer diabluras con este mando y los dos disparadores que posee, además con la posibilidad de disparar a ráfagas, hacen la operación de cazar marcianos algo muy sencillo.

Por lo que respecta al mando, al ser anatómico, permite ajustar la mano totalmente a la posición deseada. La base, redonda, es original en lo que a diseño se refiere, pero las ventosas al estar algo juntas dan un toque de inestabilidad.

La disposición de la base y la longitud de la empuñadura, complican demasiado los movimientos bruscos, aunque gracias a las ventosas esto se convierte en mal menor.

En los juegos de acción, su precisión y maniobrabilidad, lo convierten en el enemigo a batir, pero cuando hay que exigir movimientos bruscos, acabaremos más de una vez con el joystick fuera de la mesa, debido quizás a la disposición de las ventosas que al estar demasiado cerca, no hacen un buen apoyo.

La posibilidad de disparo permanente, hacen de él un arma mortífera

siblemente mejor que la otra. Lo que si podemos constatar por nuestra experiencia y lo que hemos visto, son las características más interesantes de los que hemos usado.

Lógicamente, en nuestra redacción y por nuestra mano, han circulado los más diversos *joysticks*, unos mejores que otros, unos más sólidos, otros no tanto, en suma, de características más curiosas y extrañas. Dejemos sentado que este artículo se basa en experiencias propias vividas día a día con el ordenados y estos aparatos, lo que significa que no todo lo que se escriba ha de estar de acuerdo con lo que piense el lector. Podrá considerarlo como una especie de referencia a la hora de comprar alguno, de los muchos que hay.

Para la realización de este artículo, hicimos la prueba, dedicando especial atención a los juegos de acción en diversos ordenadores, ya que al tener el *joystick* la conexión del tipo D de 9 pines, es posible conectarlo a acualquier ordenador (sea MSX o no) que tenga un port de este tipo.

Los *joysticks* se han probado con juegos muy diversos, pero la prueba final ha sido en el titulado Heavy Boxing (típico juego de acción). Se eligió este juego por varios motivos y principalmente por la rapidez de contestación que requiere un buen ataque y sobre todo una buena defensa. Otro punto a destacar es, que estos aparatos han pasado por las manos más diversas, desde el tipo tranquilo que no se altera por nada ni ante nada, hasta el nervioso que no para de disparar, aunque el juego sea un comecocos o un boxeo, como en este caso.

Y vamos a empezar como lo haría cualquier ususario. ¿Cuánto estoy dispuesto a gastarme? ¿cómo lo quiero?

Estas dos preguntas son delicadas, ya que no todos estarán dispuestos a tirar la casa por la ventana en la compra de un *joystick*. Lo ideal sería elegir uno con el que pueda realizar las más diversas operaciones, desde el diseño de dibujos hasta viajar por las galaxias más lejanas.



La primera pregunta es obvia, ya que los hay desde 2000 hasta 9000 pts., en función de lo cual podrá elegir, entre el más simple y el que se controla por infrarojos. La segunda pregunta, ya es harina de otro costal. Hay multitud de *joysticks* con la disposición de los botones de disparos más extravagantes.

Encontraremos *joystick* con un solo disparador y otros con hasta cuatro disparadores, siendo estos en teoría más completos, pero con el problema adicional del número de pulsadores que posee.

No es cuestión de buscar el más barato, ya que en la mayoría de los casos son de usar y tirar. Tampoco sirve de mucho hacerse con el más caro puesto que en estos casos el precio no está reñido con la calidad, es decir, que por costar más no va a ser mejor.

Dicen que en el término medio está la virtud y eso es lo que hay que buscar, uno que se adapte a nuestras características y necesidades.

Como no tenemos preferencias, iniciaremos la comparación.

Agradecemos al establecimieto K-Bits (C/Barquillo, 15) sin cuya colaboración no habría sido posible realizar este artículo.

TABLA DE JOYSTICKS

JOYSTICK	EMPUÑADURA	DISPARADORES	DIMENSIONES (cm) BASE	CABLE	MANDO	SUJECION	PRECIO
QUICKSHOT I	ANATOMICA	2	9x11	1,17	11	VENTOSAS	3.400
KEMPSTON	DE BOLA	2	12x9	1,20	7	GOMAS	5.500
QUICKSHOT II	ANATOMICA	2 y rafagas	12x10	1,20	12	VENTOSAS	3.500
JOYSTICK	ANATOMICA	4	13x10	1,15	11	VENTOSAS	3.700
JS-55	ANATOMICA	3	13x6	1,17	6	GOMAS	4.900
VJ-200	PLATAFORMA LISA	2	14x8	1,28	11	VENTOSAS	4.000
CHALLENGER	ANATOMICA	2 y rafagas	DIAM.13	1,25	13	VENTOSAS	2.900
VU-0001	STICK	1	12x9	1,65	4	LISO	1.800


GANE
7.000 PTAS.
todos los meses
PARTICIPANDO EN NUESTRO CONCURSO
A partir del próximo número MSX premiará mensualmente los programas que hagan llegar los lectores.
Para participar en este concurso abierto, todo aficionado a los ordenadores con este estándar deberá hacer llegar a la redacción de la revista el listado, un cassette y un texto explicativo.
Entre todos los programas que recibamos cada mes, serán seleccionados para su publicación aquellos que reúnan los siguientes criterios:
— Originalidad de la aplicación.
— Simplicidad del método de programación.
La única condición para participar en el concurso será que los programas no hayan sido publicados previamente en ninguna revista.
BANCO DE ESPAÑA
MIL pesetas
CINCO MIL pesetas

NOTICIAS



YASHICA, NUEVA OPCION EN EL MERCADO DE LOS ORDENADORES MSX

De la mano de **Dugopa, S. A.** (C/ Alcalá, 18, Madrid 28014), llega a España este nuevo ordenador MSX. Con todos los elementos necesarios para permitir codearse con otros ordenadores que llevan algo más de tiempo en la calle, este aparato puede ser la versión exótica del nuevo estándar, por lo menos hasta que veamos aquí los **JVC y Pioneer.**

El ordenador viene con los cables de adaptación a la corriente, la conexión de la antena y el que va al cassette. Además, trae un manual completo e interesante, puesto que explica todas las características del ordenador y pone al día al futuro usuario acerca del BASIC MSX. con un precio que rondará las 45.000 ó 50.000 pesetas, este nuevo MSX viene a engrosar el mercado español.

DIMension NEW EN LA CONQUISTA DEL MERCADO DEL SOFTWARE

Es inevitable hace referencia a **DIMension NEW** cuando hablamos de *software*. Actualmente, esta empresa desarrolla programas, juegos y aplicaciones para dos importantes firmas de ordenadores MSX, como son **Canon** y **Sanyo.**

Contando amplia experiencia, al haber diseñado programas para todos ordenadores, **DIMension NEW** apuesta fuerte por el estándar y parece ser que está dedicando bastantes esfuerzos en la creación de programas nuevos para el MSX.

Después del verano, se lanzarán prógramas de aplicaciones, tales como una hoja . de cálculo, un procesador de textos y un ensamblador, de los que esperamos ofrecer un completo estudios desde nuestras páginas.



EXPRESO DE ORIENTE

Este mes las noticias no pueden ser más variadas, ya que nuestro expreso, nos ha notiicado la invasión llevada a cabo por el estándar MSX en los países árabes. Gracias a la colaboración de una empresa kuwaití y a **Microsoft**, se están introduciendo ordenadores MSX en aquel país.

Con *software* árabe y el teclado tipo inglés/árabe, los primeros MSX (**Hitachi** y **Yamaha)** han interrumpido con fuerza y se espera obtener buenos resultados.

SPECTRAVIDEO, AHORA MAS BARATO

Indudablemente, alguna vez tenía que llegarle la hora a este ordenador, y parece ser que ya ha sonado. El SV-728 se comercializará, en breve, a un precio de 49.600 pesetas. no se sabe con certeza a partir de qúe fecha entrarán en vigor las nuevas tasas; pero lo cierto es que los precios bajarán.

QUICK DISK, OTRA FORMA DE GRABAR PROGRAMAS

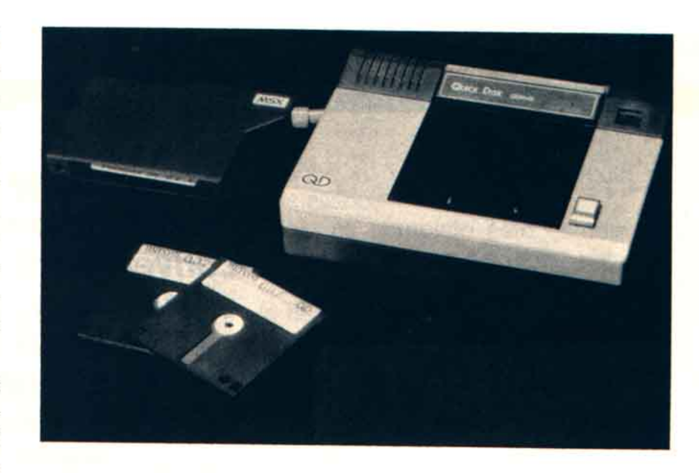
Ya anunciábamos la aparición de una nueva unidad de *diskettes*, como complemento a los ordenadores MSX. Poco ha tardado este nuevo periférico en introducirse en España.

CRESA, en Barcelona, (C/ Numancia, 107), ha tenido la culpa de ello y ofrece a los usuarios un periférico, que aunque parezca de jueguete, no lo es.

El QDM-01 es una unidad de *diskette* de 2.8 pulgadas, algo totalmente desconocido hasta el momento. Sus dimensiones, tan reducidas, lo hacen manejable y sumamente útil.

La capacidad de estos *diskettes* es de 128 K en total, divididos en 64 K por cara, ya que son de doble cara. El *interface* se conecta directamente al bus de expansión y cuenta con una fuente de alimentación que le suministra una corriente de 8.6 v. Esto es algo desconcertante, ya que, normalmente, estamos acostumbrados a ver cómo las unidades de diskettes se conectan directamente a la red.

Otra característica a destacar, es que los *diskettes* se cargan como si se tratase de un cassette; es decir, se abre una tapa, introducimos el diskette y se cierra de la misma manera.



El nuevo pariférico viene, además, con un juego de instrucciones particulares, que complementan y mejoran las ya existentes en el BASIC MSX. Algo difícil de mejorar. Estos son algunos ejemplos, QD FORMAT, QD FILES, QD KILL CASQD, etc.

La única laguna que tiene este periférico, es la imposibilidad de crear un fichero de datos para acceso aleatorio.

Resumiendo, este nuevo dispositivo reúne las cracterísticas esenciales de un *cassette* con la alta velocidad de acceso de un *diskette*, por un precio que, aunque no está definido, rondará las 40.000 pts.

TRACKBALL PARA MSX

El Marconi RB2 es un *trackball* para ordenadores MSX, con buenas características. Distribuido por **CRESA**, este periférico será de inestimable ayuda para todos aquellos que estén interesados en diseños asistidos por ordenador (CAD/CAM).

Entre las características principales, podemos destacar el soporte a la mesa mediante pequeños pies de goma, sólida carcasa de plástico, respuesta rápida a los cambios efectuados, posibilidad de sustituir a los *joysticks* y ratones en algunas tareas y tres interruptores para facilitar el trabajo al software.

Util en aplicaciones tan variadas como el Tratamiento de Textos, diseños asistidos CAD/CAM y en juegos, este periférico puede dar mucho que hablar.



HABLEMOS DE

o de como conseguir que no

No hay nadie que no se sienta sorprendido de no entender nada cuando escucha una conversación en el autobús que suena a castellano y, sin embargo no parece serlo.
—"Tengo un micro (microordenador) muy potente, tienen cepeeme (CP/M) concurrente, cientoveintiocho cas ram (128k RAM), interfeis serie y paralelo (INTERFACE CENTRONICS Y RS–232), además de un maus (ratón) y dos draibs (drives) de cinco y cuatro (5.25")."



ORDENADORES

nos entienda nadie

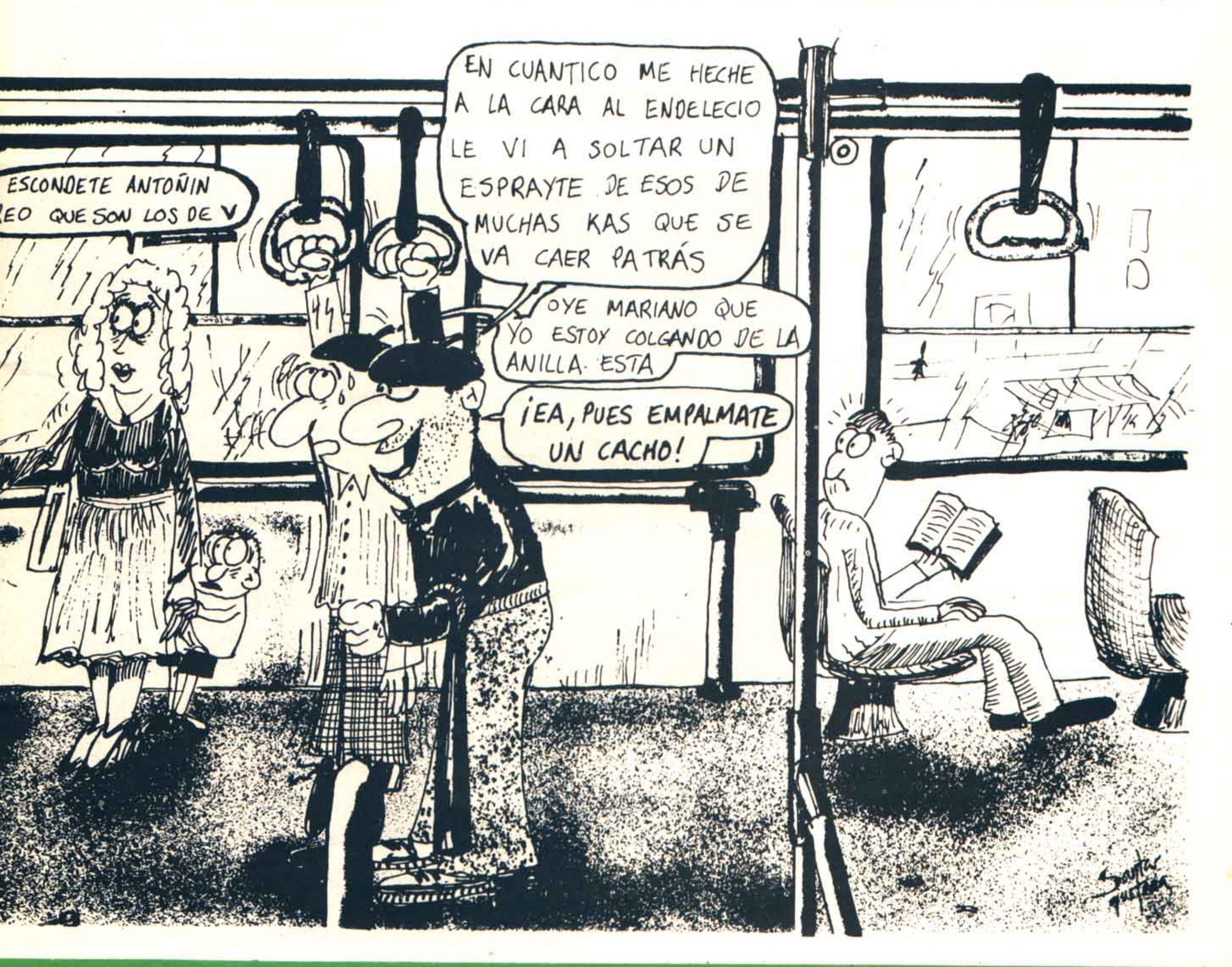
Entre paréntesis podemos hallar las denominaciones más habituales de estas cosas a la que se refiere en su jerga el orador. Si somos, como normalmente suele ocurrir, unos aprendices del campo de la informática, todo este guirigay nos habrá sonado a chino, y podemos llegar a la conclusión de que el que así habla es un Gran Maestro en el Esotérico Mundo de la Informática, algo así como un Sumo Sacerdote. Pues no, es altamente probable que nos equivoquemos y no que no sea más que un Ilustre Engañado por la Alta Terminología de la impresionante Sonoridad.

Ante tamañas confusiones, la dirección de esta revista ha decidido encender un gran Faro en el Ténebre Mundo del Habla, y concedere a los lectores la llama de la Buena y Castella Expresión. Para ello, he elaborado el siguiente "DICCIONARIO ESPANGLISH–ESPAÑOL", compendio de los términos de uso más frecuente en inglés, su fea versión en espanglish y su versión más correcta en castellano.

No es completo este estudio, ya que hay muchas de estas palabras que no tienen traducción directa, y si la tienen suenan aún peor que el original.

De la memoria: olvidemos el castellano

Uno de los puntos cruaciales en un ordenador, es la memoria de que



HABLEMOS DE

dispone. Pues bien, aquí es donde comenzamos a fracasar miserablemente en nuestro intento de hablar bien. La memoria se divide en dos bloques principales RAM y ROM (pronunciadas tal cual se escriben). La primera es la que el usuario tiene a su disposición para equivocarse, y la segunda es aquella en la que se equivocó el fabricante (definición sacada de la experiencia propia y ajena: todoos se quejan de los fallos de los ordenadores, y nadie hace nada completamente a derechas).

En castellano, sus nombres son:

> *RAM*=Memoria de Acceso Aleatorio (MAA)
> *ROM*=Memoria de Sólo Lectura (MSL)

Como nadie nos entendería si hablásemos de la MAA o de la MSL, nos vemos obligados, a nuestro pesar, a recomendar que se hable de memoria *RAM* y *ROM*.

Otro punto a estudiar es las agrupaciones de memoria. En el punto más elemental tenemos el *bit*, que es un tonto que dice 'si' o 'no' (nota: la definición de ordenador es: Un millón de tontos diciendo SI y NO muy rápido), lo que clásicamente se dice '1' o '0. Su nombre proviene de las palabras inglesas *Binary Digit*=Digito Binario, lo que nos lleva a sugerir como nombre castellano DIO. Como siempre, nadie nos hará caso y le seguirá llamando BIT.

La siguiente unidad es el *NIBBLE*, palabra intraducible que corresponde de 4 *bits*. Se pronuncia 'nibel', y así habremos de llamarla, mal que nos pese. Siguiendo nuestra escalada, nos encontramos con el *'byte'*, 8 *bits*, ponunciado bait. Su traducción es MORDISCO, que todavía nadie ha aceptado como denominación española, a pesar de ser tan cómica como la inglesa (sólo que comprensible). En cambio, se emplea habitualmente una denominación tan anodina como OCTETO.

Otra unidad, menos usada, es la *'word'*, normalmente (no siempre) de 2 bytes. La traducción directa, que es de hecho la que se usa es PALABRA.

En términos mayores, se habla de *Kbytes*, o simplemente K, que es un conjunto de 1024 octetos. la denominación mejor sería KILOOCTETOS, pero, por ser muy larga, damos la alternativa de usar el término 'K' (pronunciando CAS). Cuando uno utiliza aparatos de mayor volumen, pueden hablar de los MEGAS y los GIGAS (1024K y 1024Gigas, respectivamente), que, por tratarse de prefijos griegos, admitiremos como válidos.

Este apartado, como hemos podido observar, no ha sido un gran avance, ya que no hemos castellanizado casi ninguna palabra, pero os garantizamos que en apartados posteriores, daremos pasos de importancia.

Los periféricos: como hablan los ordenadores

En un primer nivel, el ordenador habla con nosotros a través de una pantalla–televisor–monitor y de un teclado. Como se puede ver, hasta aquí no se presenta ningún problema de terminología, todo es buen castellano. Pero a veces nos hablarán del CRT, que es la denominación común a televisión y monitor, que en castellano se llama Tubo de Rayos Catódicos o, más abreviadamente, TRC. Otra posibilidad es el *LED–Display*, en castellano la pantalla–LED, o bien el LCD–Display, es decir, la pantalla de cristal líquido.



Escuela e Informática, dos mundos unidos por el afán de superación, y coordinados en el presente, como base real del futuro próximo.

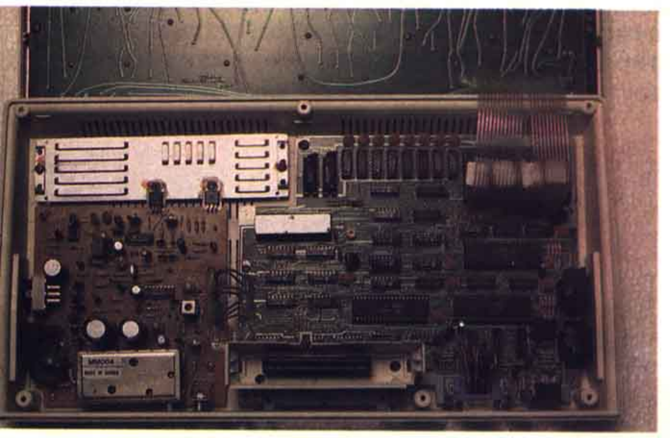
ORDENADORES

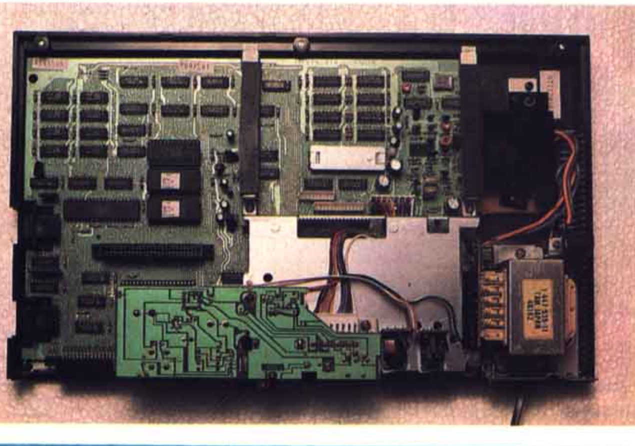
Como alternativa al teclado tenemos el *'mouse'* (RATON), la bola gráfica el *'joystick'* (MANDO DE JUEGOS) y el tablero gráfico, todos ellos con usos muy específicos.

Cuando queremos enchufar una impresora a nuestro ordenador, nos enfrentamos a la alternativa que nos presenta el vendedor: ¿La quiere con *CENTRONIC o RS-232? Esto que nos dice el vendedor no son dos marcas de impresoras, son dos modos de hablar el ordenador con la impresora. Al primero, por comodidad, le llamaremos transmisión paralelo y al segundo transmisión serie. Estos nombres son generalizadores, ya que transmisión paralelo también es el sistema IEEE–488 y paralelo el RS–422, pero como no son muy habituales los dejamos de lado. El sistema paralelo, lo que hace es envitar un bit* por cable, por los que envía un octeto cada vez, lo que es rápido pero caro al exigir 8 cables para la transmisión. En cambio, el sistema serie, envía los 8 *BITS* uno detrás de otro, por un sólo cable, lo que lo hace más barato pero más lento. De todos modos, no se ha de elegir el sistema por economía, sino por el ordenador que se tenga. En el caso del MSX, el sistema que tiene implementado es *CENTRONICS*, es decir, paralelo.

Otros periféricos de interés son los discos, de los que hay varias categorías:

DISCOS FLEXIBLES: Mal llamados *FLOPPY DISCS*, los hay en tamaños de 5,25 pulgadas y 12 pulgadas.
DISCOS CERAMICOS: En tamaños de 3, 3.25, 3.5, 3.9 pulgadas, el utilizado por los MSX es el de 3.5 pulgadas, que es el tamaño que se está imponiendo en todo el mundo.
DISCOS DUROS: También llamados (mal) *HARD DISKS* o *WINCHESTERS*, sirven para almacenar mucho y muy rápido, y sólo tienen utilidad real para grandes sistemas, oficinas y programadores profesionales.





El ordenador visto por dentro, aparece ante nosotros como un laberíntico mundo de entramados y chips en continua actividad. Su progreso rompe todas las fronteras económicas y sociales hasta las metas más insospechadas. El desarrollo de sus sistemas es nuestro futuro.

En general, a los discos flexibles y cerámicos se les llama *DISKETTES*, lo que es vulgar y malsonantes hasta en inglés. Como nombre corto para ellos, proponemos el que se usa desde siempre entre profesionales, DISCOS, ya que a los discos duros se les da el nombre completo. No hay confusión posible.

Existen muchos otros periféricos, pero su uso es menos habitual y, por tanto, aparecen mucho menos en las conversaciones.

Dentro de este apartado cabe tratar también de la terminología técnica que se utiliza (por suerte, pocas veces) para describir la interacción entre el periférico y el microprocesador, en sus niveles más íntimos.

Por ejemplo, se habla de *'handshake'* ("jancheik"). Esto significa APRETON DE MANOS, y quiere decir el intercambio de señales o controles que realizan periférico y procesador para poder llegar a un traspaso de datos eficiente. Viene a ser una especie de:

—"Oye, que te envío un bloque de datos"
—"Un momento, que ahora no te puede hacer caso"
—"Vale, espero"
...
—"Chico, que ya puedo recibir"
—"Bueno, pues cuando suene la tercera señal, te los envío"
...

Esto, que es imprencindible (hablando se entiende la gente), el *'handshake'*, y en castellano se utiliza para denominarlo la palabra PROTOCOLO, por su gran parecido con

HABLEMOS DE

las ceremonias diplomáticas y cortesanas.

El BASIC: traducir y no adaptar

Hay muchos bárbaros sueltos entre la gran masa de aficionados a los ordenadores que son capaces de destrozar el castellano hasta su misma fuente. Allá donde es muy sencillo decir, por ejemplo, IMPRIMIR, los muy Atilas dirán *PRINTER*. Este es uno sólo de los múltiples ejemplos que podemos hallar. Como casi todos ellos se producen en usuarios de BASIC, y son fruto de la adaptación de instrucciones inglesas a una nomenclatura con una cierta sonoridad castellana.

Aparte de la burrada ya mencionada, cabe denotar las siguientes, todas ellas reales (soy testigo):

IMPUTEAR por PREGUNTAR o PEDIR
FORMATEAR por DAR FORMATO
PLOPEAR por PINTAR (UN PUNTO)
GOSUBEAR por LLAMAR A UNA SUBRUTINA
ERASEAR por BORRAR
etc...

Mi recomendación es que, en caso de duda o de que sea muy larga la traducción en castellano, se utilice el original inglés sin modificción, pues por lo menos se tienen garantías de que es una palabra válidad y existente en algún idioma.

Un viaje al interior (I): El soporte lógico

Supongo que la primera duda que aparece ante este título es "¿Qué es eso de Soporte Lógico?". Pues bien, el soporte lógico es lo que vulgarmente se llama *"sofware"* (pronunciado 'sofgüer'). El título por sí es ya una declaración de principios sobre lo que queremos conseguir con este artículo: hablar español.

Dentro del soporte lógico, cabe destacar tres estructuras a analizar:

—El sistema operativo
—Los lenguajes
—Los programas

El sistema operativo es aquella parte que viene con el ordenador y que sirve para su comunicación con el mundo exterior a través de los periféricos. En el caso del MSX se denomina MS–DOS.

El sistema operativo de un ordenador incluye una serie de rutinas para conseguir realizar tareas como grabar en disco o cinta, recuperar datos de ellos, dibujar en pantalla o impresora, hacer ruidos, etc.

Los lenguajes son lo que nos permiten hacer con el ordenador lo que nos apetezca (siempre que sepamos cómo). Estos lenguajes tienen tres variantes importantes, a saber: el código de máquina, los intérpretes y los compiladores.

El código de máquina es un conjunto de unos y ceros que sirven para que el microprocesdor ejecute una serie de acciones sencillas (saltar, sumar, complementar o transferir datos) y que es, de hecho, el único lenguaje que entiende.

Un intérprete lo que hace es leer un programa escrito en un lenguaje distinto del código de máquina, instrucción a instrucción y efectuar una serie de operaciones en código de máquina (u otro lenguaje) para realizar la acción que se espera de él. Tal es el caso del intérprete de BASIC que lleva incorporado un ordenador MSX.

Un compilador lo que hace es leer completo un programa escrito en algún lenguaje y traducirlo como conjunto a una serie de opraciones en código máquina. No olvidemos nunca que los ensambladores (traductores de lenguaje ensamblador a código de máquina) son los más sencillos de los copiladores.



El ordenador, señor de la galaxia informática, al servicio de nuestros quehaceres y centro de nuestras preocupaciones.

ORDENADORES

Para aclarar las diferencias entre intérprete y compilador, supongamos el caso de un conferencia que se diese a gente de muy diversos países. Para hacerles comprender a todos lo que se quiere decir, se optaría por una de dos alternativas:

a) Darles la conferencia ya traducida, con lo que sólo habrían de preocuparse de leerla.

Esto es un compilador.

b) Poner un traductor simultáneo. Esto es lo que hace un intérprete, traducir simultáneamente.

Como última diferencia, destacaremos que un programa suele ejecutarse más lentamente al interpretarse que al compilarse.

Los programas se dividen en Aplicaciones y Juegos (principalmente). Las aplicaciones son cosas como procesadores de textos, bases de datos, hojas de cálculo (también llamados estadillos u hojas electrónicas)...Los juegos los conocen todos, por lo que no exigen más explicación.

Viaje al interior (II): El soporte físico

Del mismo modo que antes hablábamos del soporte lógico, el soporte físico es el *'Hardware'* (pronunciación: 'jargüer'), y que consta de todo lo que tiene en realidad tangible dentro del ordenador.

Como elemento fundamental, aparece el microprocesador, un bicho habitualmente negro grisáceo con muchas patas (en el MSX, se llama Z-80 y tiene 40 patas). Este es el corazón del aparato, el que ordena y manda, siempre bajo tus órdenes..

Junto a él se hallan muchos otros bichos del mismo color, y algunas patas menos que son, en el caso del MSX, la memoria ROM, la memoria RAM, el Generador Programable de Sonidos AY-3-8910, el generador de Vídeo 9816 y algunos dispositivos lógicos. Todos ellos sirven para que el microprocesador pueda hacer algo útil.

El microprocesador se comunica con los arriba mencionados a través de una cosa llamada 'buses', y cuya más correcta traducción al castellano aparecida hasta ahora es, precisamente, 'buses'.

Estos buses con conjuntos de hilos por lo que viajan unos y ceros (por eso se les llama buses, porque por ellos viajan cosas) y que cumplen muy diversas funciones, de las que depende el nombre que se le dé al bus por el que van.

En primer lugar, tenemos el Bus de Datos, que sirve para transferir datos entre el procesador y la memoria y periféricos. Para controlar este flujo, tanto su dirección como su significado, se utiliza el Bus Control. Finalmente, para controlar su destino, se utiliza el Bus de Direcciones.

Valga esta corta perorata para introducir a la vez la nomenclatura castellana y los fundamentos de operación del interior de un ordenador.



Paquete de software. El mercado camina hacia la consolidación económica de forma espectacular, con desprecio absoluto de la crisis.

Despedida

No podemos negar que se nos han quedado algunas palabras de uso más o menos frecuente por traducir, pero son, en general, vocablos que no utilizamos normalmente en nuestra charla habitual. Por ello, para dichas palabras, remitimos a un diccionario inglés-castellano que, si no nos dicen su traducción plenamente correcta, al menos nos da buenas ideas sobre lo que significa y cómo podríamos decirlo.

Como última recomendación, recordar que estamos en España y ello hace recomendable hablar en español. Pero recordad siempre: SI QUEREIS DESPISTAR O ASOMBRAR A VUESTROS AMIGOS, UTILIZAD TERMINOLOGIA INGLESA Y, SI ES POSIBLE, PRONUNCIADLA MAL. Esto último es a lo que se han dedicado durante el último decenio aquellos que han querido dárselas de "enteraos".

LAS COMUNICACIONES EN LOS ORD

EL INTERFACE

La necesidad de conectar un ordenador con el exterior se pone de manifiesto cuando se desea, por ejemplo, volcar un fichero de texto a la impresora, por no hablar de la transmisión de ficheros de un ordenador a otro através de *modems* para la línea telefónica.

En ambos casos se trata de transmisión de datos, aunque la conexión con la impresora no sea un ejemplo específico.

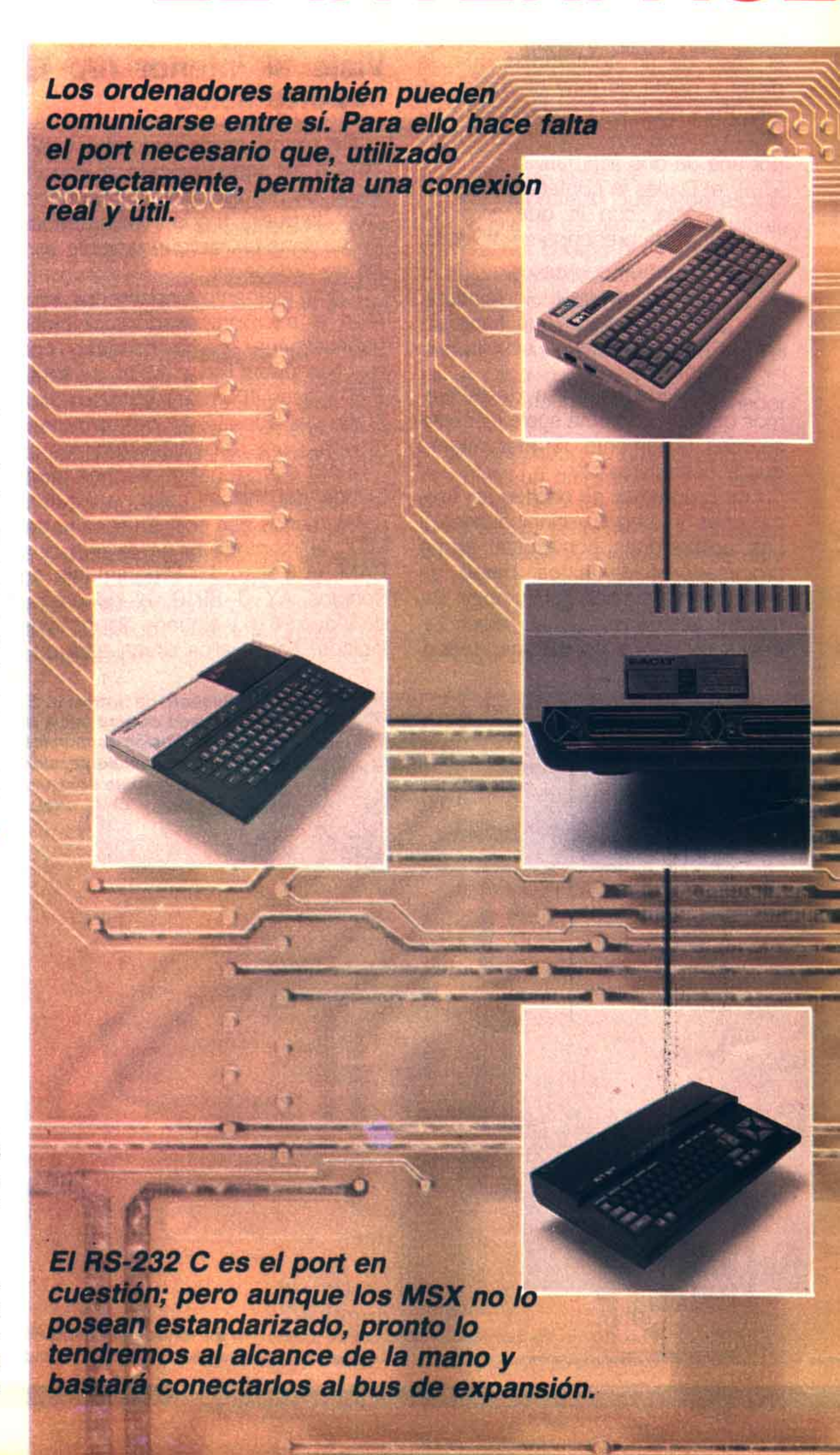
Generalmente, la idea de transmisión de datos se asocia, en el ámbito del ordenador doméstico, conectado al acoplador acústico y éste al teléfono. No tiene porque ser así en otros entornos, en los que se efectúa la conexión entre ordenadores para formar una red local **(LAN – Local Area Network)** mediante cable coaxial o fibra óptica, o en la conexión de terminales inteligentes a otros terminales u ordenadores mediante *modem* a través de línea telefónica dedicada, etc. Hoy en día puede afirmarse que las comunicaciones entre ordenadores son, junto a las tecnologías de fabricación de circuitos integrados, los dos pilares más importantes de la industria del ordenador.

La conexión entre equipos para la transmisión de datos, se puede realizar de dos manera: en serie y paralelo.

La conexión entre dos equipos para la transmisión de información, se puede realizar de dos maneras diferentes, que pueden ser:

- Transmisión paralelo o serie.

La conexión paralelo tiene la particularidad de que todos los *bits* se transmiten al mismo tiempo. Ello implica que si tenemos un formato de palabra de 8 *bits*, harán falta como



Los ordenadores también pueden comunicarse entre sí. Para ello hace falta el port necesario que, utilizado correctamente, permita una conexión real y útil.

El RS-232 C es el port en cuestión; pero aunque los MSX no lo posean estandarizado, pronto lo tendremos al alcance de la mano y bastará conectarlos al bus de expansión.

ENADORES PERSONALES: RS-232 C.

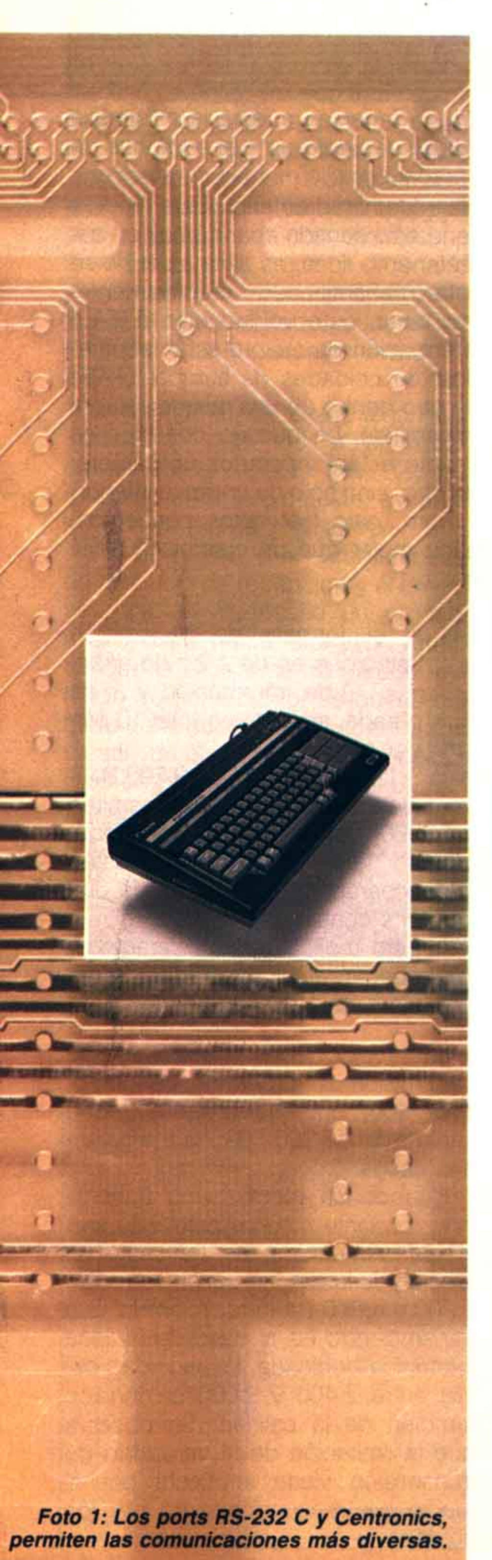


Foto 1: Los ports RS-232 C y Centronics, permiten las comunicaciones más diversas.

mínimo 9 hilos: los 8 anteriores más la masa. Si tenemos una impresora situada a 6 metros del ordenador, tendremos que efectuar la conexión con 9 hilos de 6 metros cada uno. Si la distancia es mayor, habrá que intercalar un regenerador de señal, que son repetidores digitales, colocando 8 regeneradores. El coste de la transmisión en paralelo se dispara cuando la distancia supera los 15 metros. Por tanto, este tipo de transmisión se emplea, preferentemente, den distancias costas de hasta 15 metros, para conexión de periféricos, como impresoras, unidades adicionales de disco, aparatos de medidas, etc.

El interface Centronics

El *interface* más conocido es, seguramente, el **Centronics** (que además incorpora el MSX), aunque industrialmente hay otros *interface* paralelo, orientados a la conexión de equipos de medidad automáticas (polímetros, frecuencímetros, fuentes de alimentación, etc), como la IEEE–488.

La conexión serie, por el contrario, necesita sólo dos hilos como mínimo, uno para los datos y otro para la masa. En este caso, los *bits* de la palabra del ordenador se mandan uno tras otro. La ventaja principal es el menor coste.

Hay que distiguir dos maneras de transmitir datos por una línea, de forma síncrona y asíncrona. En el primer caso, tanto el receptor como el emisor están sincronizados, por lo que se pueden mandar en un principio los 8 *bits* de una palabra a continuación de los 8 *bits* de la otra. Al estar los equipos sincronizados se reconoce perfectamente cuando empieza y cuando acaba una palabra de 8 *bits*. Este modo de transmisión serie se utiliza, de manera más sofisticada, en comunicaciones de alta velocidad.

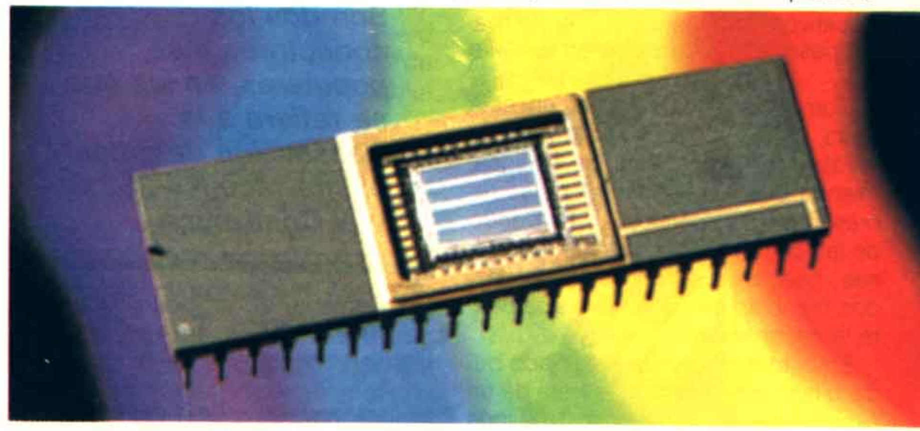


Foto 2: Los chips son elementos indispensables para una buena transmisión de datos.

La transmisión asíncrona, por el contrario, es menos costosa. Cada palabra va empaquetada entre los bits de arranque y de parada. De esta forma, al detectar un *bits* de arranque, el ordenador sabe que detrás viene una palabra que termina con la detección del *bit* de parada. En la figura 1 se ilustra este modo de transmisión serie.

El hecho de que cada palabra vaya empaquetada, significa que la sincronización de los equipos se efectúa con cada ordenador. En la figura 1 se aprecia que la separación entre palabras no es uniforme, porque tampoco es necesario, debido a que cada palabra tiene su propia sin-

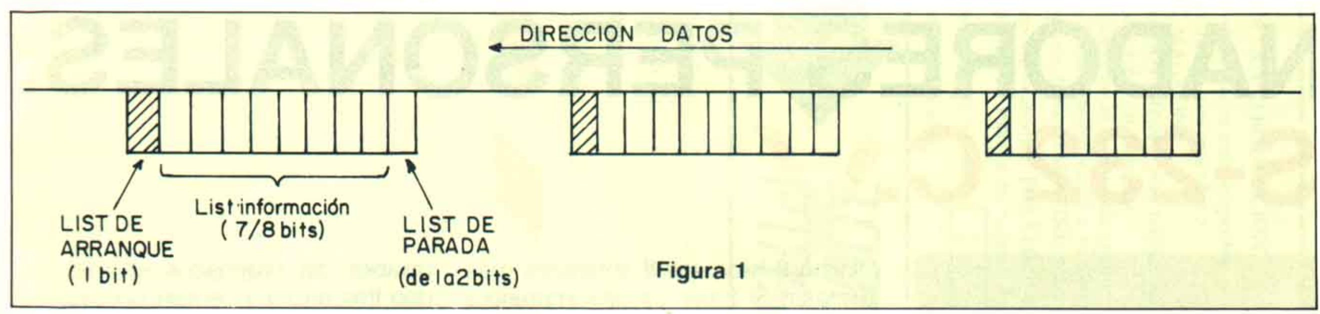


Figura 1

cronización. No sucede lo mismo en el modo de transmisión especial, llamada trama de sincronismo, cuando no hay caracteres que enviar y así, mantener los equipos sincronizados. Esto hace que, generalmente, sean más complicados.

Volviendo a la transmisión asíncrona, hay que señalar que si, por ejemplo, queremos enviar un *byte*, el equipos o *interface* de transmisión va a enviar primeros los bits menos significativos, terminando por el de mayor peso, como se indica en la figura 2.

Como podemos apreciar en esta figura. los unos lógicos se representan mediante estados bajo de la línea, mientra que los ceros lógicos se representan con niveles altos. Dentro de las comunicaciones serie asíncronas. pueden distinguirse. a su vez, dos maneras de transmitir físicamente la información.

En primer lugar está el llamado bucle o lazo de corriente. Se trata de un circuito que se cierra en los equipos de transmisión, circulando una corriente de aproximadamente 20 mA. Cuando el circuito se abre, no circula corriente, por los que es posible distinguir los dos estados de un bit. Este bucle de corriente se emplea en los teletipos (TELEX). en segundo lugar, se tiene la transmisión por niveles de tensión. Esta es la más extendida en los ordenadores personales. Se trata de la norma RS-232 C. En ella, un nivel lógico alto, es decir, un uno, se representa por una tensión entre −3 y −25 v., y un cero lógico, por una tensión entre +3 y +25 v.

Son dos los conectores más populares, en los que se refiere a la transmisión de datos: el interface RS-232 C y el Contronics.

Normalmente se emplean tensiones de −9 a −12 v. y de +9 y +12 v., habiendo demostrado la práctica, que tensiones por debajo de 9 v. absolutos no son adecuados a pesar de cumplir la norma.

Otro parámetro a tener en cuenta es la velocidad de transmisión. Esta se mide en Baudios. No se debe efectuar el cálculo en *bits* por segundo sin tener en cuenta la existencia del *bit* de arranque y él o los *bits* de parada, por ello es más cómodo emplear la unidad anterior. Esta medida tiene sólo sentido cuando hay un carácter en la línea, es decir, cuando se está transmitiendo efectivamente. Por tanto, esta unidad sólo nos da información de cuánto están separados, en unidades de tiempo, un *bit* de otro dentro de una misma palabra transmitida. Lo que no nos dice es cuánto están separados un carácter de otro, sino sólo da un límite inferior.

Pero, para fijar estos conceptos, nada mejor que un ejemplo numérico:

— si el formato de transmisión asíncrona es de 1 *bit* de arranque, 8 de información y 1 de parada, se tiene en total 10 *bits* por carácter.
— Si la velocidad es de 9.600 Baudios, resulta que se transmiten 960 caracteres por segundo, por tanto en un carácter están separados 1/9600 seg. de dos *bits* consecutivos.
— Esto implica que dos caracteres puedan estar separados un mínimo de 1/960 de segundo.

Las velocidades habituales de transmisión asíncrona van desde 75, 110, 300, 600, 1200, 2400, 4800 hasta más de 9600 Baudios en algunos casos.

Cuando se efectúa una transmisión mediante *modem* para línea telefónica conmutada, se transmite en función de la calidad del enlace entre 1.200 y 4.800 Baudios, mientras que transmitiendo en el caso de la línea telefónica dedicada, la velocidad oscila entre 2.400 y 9.600 en función también de la calidad. Se observa que la limitación de la velocidad de transmisión viene impuesta por el medio de transmisión y no por los equipos.

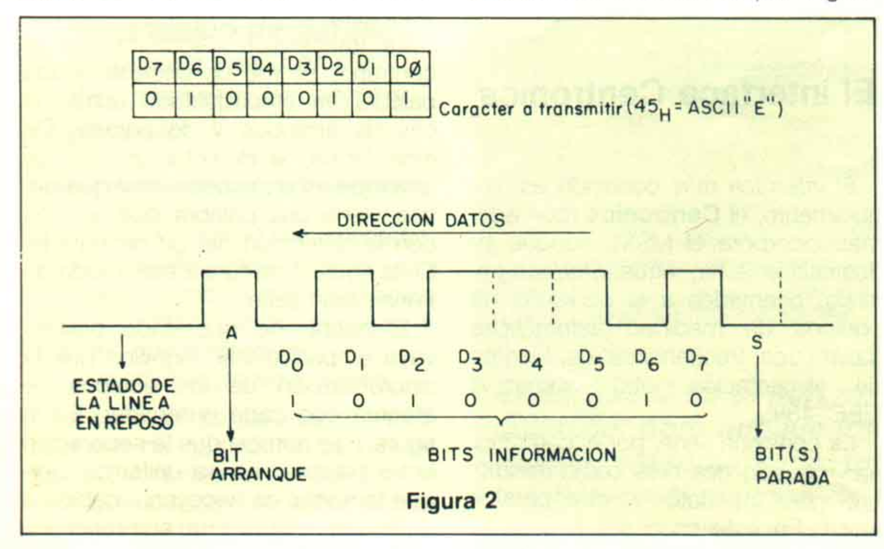


Figura 2

En la transmisión asíncrona serie, hay que tener en cuenta la rapidez con la que el ordenador es capaz de almacenar el carácter recibido. Si se ha transmitido un caracter y el ordenador está realizando tareas, si trasmitimos otro carácter, se perderá el carácter anterior. Si el ordenador tarda en darse cuenta de que ha recibido un carácter o si tarda al almacenar ese carácter recibido, más que el tiempo mínimo entre caracteres para la velocidad de transmisión fijada, también se perderá el carácter en el caso de transmitir caracteres seguidos, uno detrás de otro, sin interrupción. Esto significa que un ordenador puede tener un *interface* RS-232 C a 9.600 Baudios, debido a que el mecanismo de atención al *interface* RS-232 C es lento.

Se ve que es necesario algún procedimiento para indicar al equipo emisor que el receptor está dispuesto para recibir información, evitando así que se pierda información.

Por otra parte, si el ordenador está en un bucle de recepción y el emisor se para, es necesario un procedimiento para indicar al equipo receptor que el emisor se ha parado, evitando que el receptor se quede colgado en el bucle de recepción y pueda salir de él. Se trata de las líneas o hilos de protocolo (Handshake).

Antes de describirlas, es preciso conocer algo más sobre la norma RS-232 C. Además de especificar los niveles de tensión de las señales, fija la norma y tipo de conector para este *interface* y la asignación de los diversos *pines* del conector a diferentes funciones.

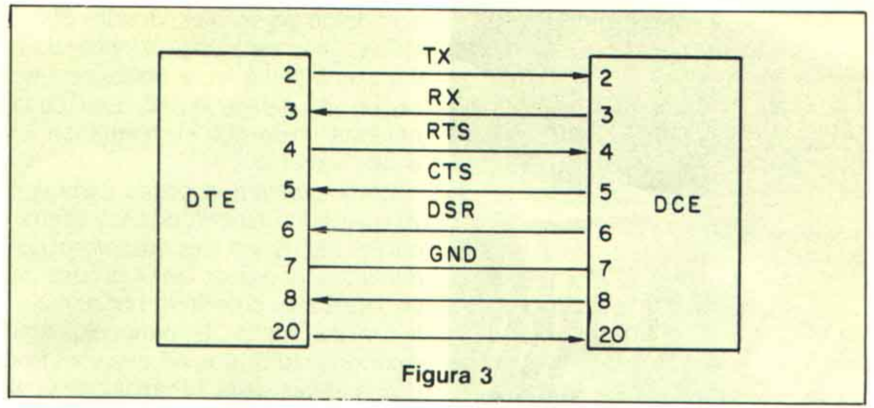


Figura 3

El interface RS-232 C

El conector es un subminiatura D de 25 *pines*.

Ciertamente hay fabricantes que no emplean este conector, usando bien uno con menos número de *pines* o, incluso, uo con mayor número de ellos, en función del número de ellos, en función del número de *pines*, no existe posibilidad de confusión.

La asignación de los *pines* depende de la fabricación de nuestro equipo. Puede tratarse de un equipo terminal de datos (DTE), en el que la información entra o sale o de un equipo de comunicación de datos (DEC), que permite el paso de la información a través suya.

En el primer caso, se tienen los terminales y los ordenadores principales (Host) mientras que en el segundo están los *modems* y ordena-



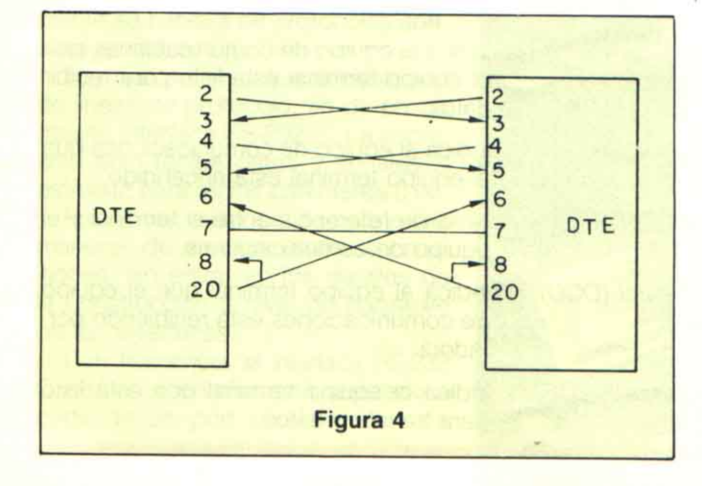


Figura 4

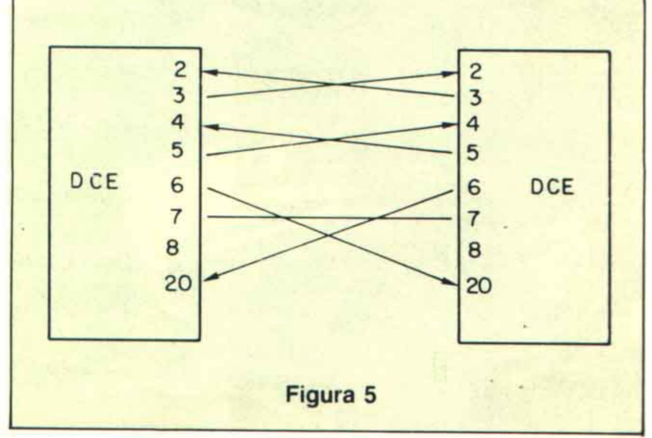


Figura 5

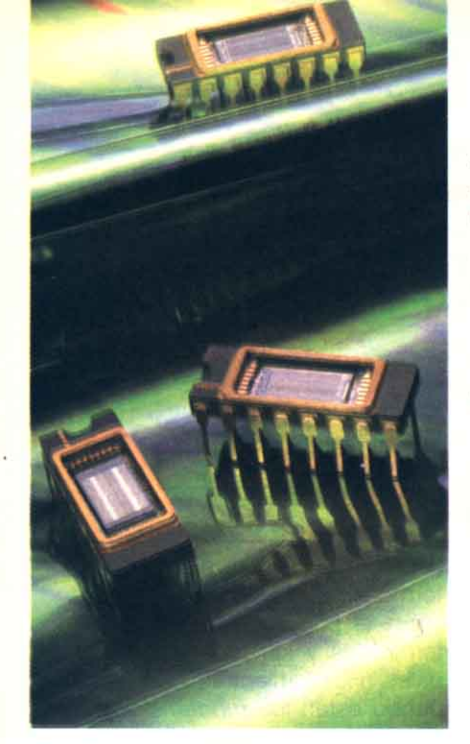


Foto 3: Reducir el tamaño de los chips, es una forma de agilizar y elevar la velocidad de transmisión.

dores de comunicaciones. Puede deducirse que al estar incluído el término ordenador en ambos grupos, estos podrán ejercer ambas funciones según el equipamiento de que dispongan.

La diferencia, a nivel de *pines* del conector, estriba en la inversión del sentido de las señales, donde habían salidas hay entradas y viceversa. Esto tiene como fin, el permitir la utilización de cables en los que cada *pin* está unido con su homólogo en el otro extremo.

En la práctica ls cosas cambian, ya que, al no tener todos los *interface* RS-232 C los mismos *pines* conectados, pudiendo darse el caso de un fabricante prescinda por completo de las líneas de protocolo, empleando solo 3 hilos en el cable. Por ello, además de la información suministrada por el fabricante, suelen suministrar los cables específicos para comunicar su máquina con otra determinada.

Si uno desea construir su propio cable, deberá poder determinar previamente qué *pines* son salida y cuáles entrada en los equipos a conectar. De esta forma, al menos no habrá problemas destructivos.

Antes de proseguir con la construcción de cables, es interesante ver la asignación de pines:

ASIGNACION PINES

PIN	Nombre de la señal	Función
1	*Protective Ground*	Toma de tierra del equipo (carcasa).
2	*Transmitted Data* (XT)	Salida de datos transmitidos del equipo terminal al equipo de comunicaciones.
3	*Recieved Data* (RX)	Entrada de datos recibidos por el equipo terminal.
4	*Request to send* (RTS)	Indica al equipo de comunicaciones que el equipo terminal está listo para transmitir datos.
5	*Clar to send* (CTS)	Indica al equipo de comunicaciones que el equipo terminal está listo para recibir datos.
6	*Data Set Ready* (DSR)	Indica al equipo de comunicaciones que el equipo terminal está encendido.
7	*Signal Ground* (GND)	Nivel de referencia entre el terminal y el equipo de comunicaciones.
8	*Data Carrier Detect (DCD)*	Indica al equipo terminal que el equipo de comunicaciones está recibiendo portadora.
20	*Data Terminal Ready(DTR)*	Indica al equipo terminal que está listo para transferir datos.

Existen más señales; pero se omite en esta descripción por ser poco frecuente su implementación. Puede observarse que la tabla anterior indica como conectar un equipo con otro, según sea un terminal o un equipo de comunicaciones. En un caso los *pines* son entradas y en otro salidas. Por ello es imprescindible saber si el ordenador está configurado como equipo terminal (DTE) o como equipo de comunicaciones (DTE).

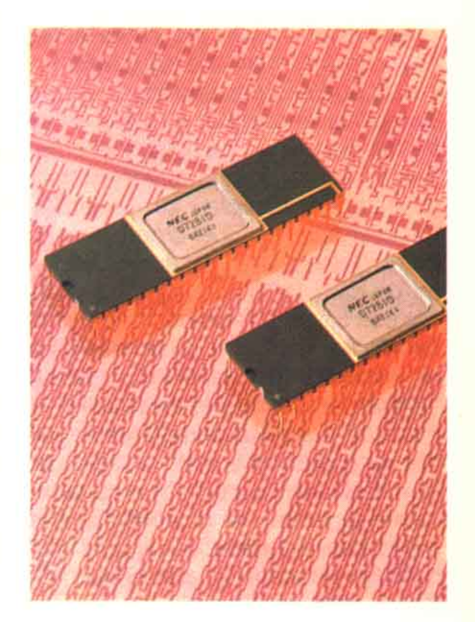


Foto 4: VLSI (Very Large Scale Integration — Integración a escala muy grande), optimizan el rendimiento.



Foto 5: La transmisión de datos, la informática y el hogar, fundidos en una sola aplicación.

En la siguiente figura, se ilustra la conexión de dos equipos en función de su configuración. mostrando claramente los *pines* que son salidas y los que actúan como entradas:

(Figura 3)

Si se desea conectar dos equipos terminales entre sí, el cableado será:

(Figura 4)

Ypara dos equipos de comunicaciones:

(Figura 5)

Es posible que los *interface* RS-232 C, nos dispongan de todas las señales descritas anteriormente. En este caso habrá que ver las señales de un equipo y las de otro, generando las señales que falten mediante puentes en el conector.

Como existen muchas posibilidades, en la siguiente figura, se muestran sólo dos casos, esperando que el lector pueda construir su propio cable con ayuda de este artículo.

En el caso de contar con un equipo, que no dispongan de las líneas de protocolo (CTS, RTS, DSR, DTR), ni de la línea auxiliar DCD y quiera conectar con un equipo, que disponga de algunas de estas líneas, tendremos una conexión a 3 hilos, como queda reflejada en las figuras 6 y 7.

En el primer caso, es evidente que el ordenador que tiene el *interface* sin líneas de protocolo, debe ser suficientemente rápido para no perder caracteres en recepción, y el que dispone de ellos también debe ser rápido, ya que no existe ninguna manera de indicar al contrario que se espero, al menos a nivel de líneas de *interface* RS-232 C aunque posteriormente se hablará de protocolos *software* sencillos.

Hay que señalar, que la existencia de líneas de protocolo, se da en los demás *interface* RS-232 C. La impresora sólo recibe datos y señales si está lista para recibir caracteres o no.

Por ahora, sólo se han visto las maneras de efectuar distintas conexiones, sin entrar en los detalles de la construcción a nivel de *hardware* de los *inyerfaces*.

Con frecuencia, el interface RS-232 C de un ordenador está realizado a partir de un *port* paralelo. En este caso existe una limitación en la velocidad de transmisión y recepción, al ser realizadas estas funciones mediante algoritmos *software*.

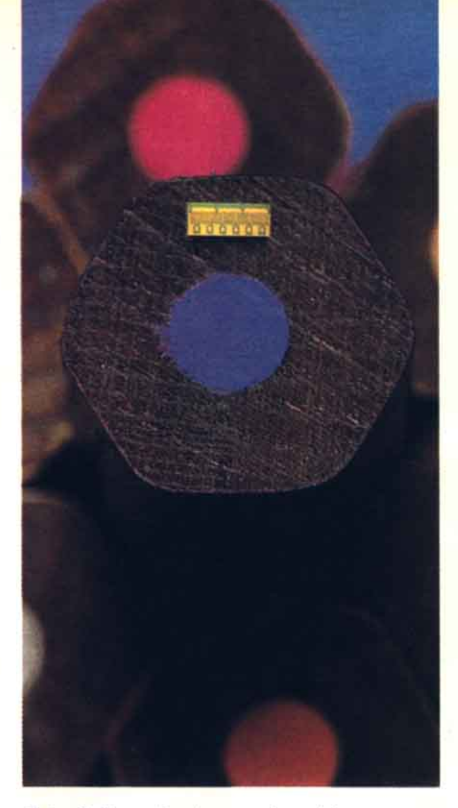
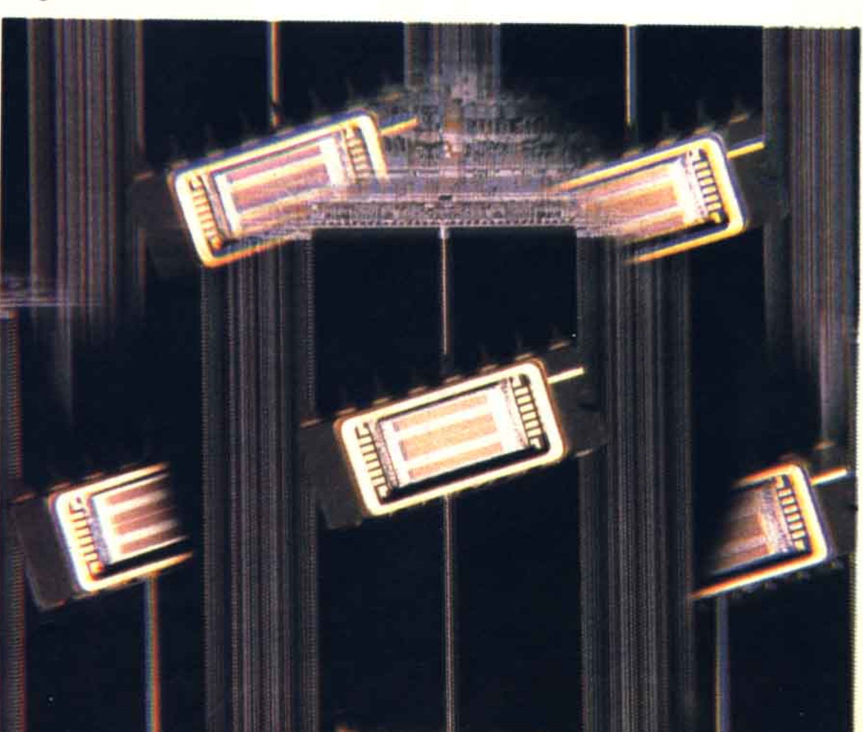


Foto 6: Tamaño de un microchip comparado con la mina de un lápiz.

Otra limitación es el hecho de solo poder realizar una función a la vez, o bien transmitir o bien recibir, sin un solo retraso muy grande o lo que es lo mismo, a una velocidad muy baja.

Foto 7: En un futuro, los chips realizarán las funciones más inverosímiles, que uno pueda imaginar.



Se dice que la conexión es semiduplex, transmitiendo en un instante y recibiendo en otro.

Existen por el contrario circuitos integrados específicos para transmisión asíncrona, llamdos UART (*Universal Asynchronous Reciever-Transmitter*), siendo una solución más elegante que la anterior. El *software* de atención al *interface* es más reducido, al ocupar el circuito integrado de las funciones específicas de serialización y control. Permite además, simultanear la transmisión y recepción de datos, estableciendo una conexión duplex.

El control de las líneas de protocolo (CTS, RTS) se realiza automáticamente, pudiéndo además, leer el estado de las líneas DSR y DCD y manejar mediante el registro adecuado, la línea DTR entre otros. Existen también circuitos integrados con dos*interfaces* serie asíncronos, llamados DART o DUART, que permiten la realización de equipos de comunicaciones (DCE) con pocos circuitos integrados. El empleo de estos UART permite un uso más universal del *interface*, al poder realizar cambios en los diversos parámetros de la transmisión de manera sencilla, programando simplemente el registro adecuado.

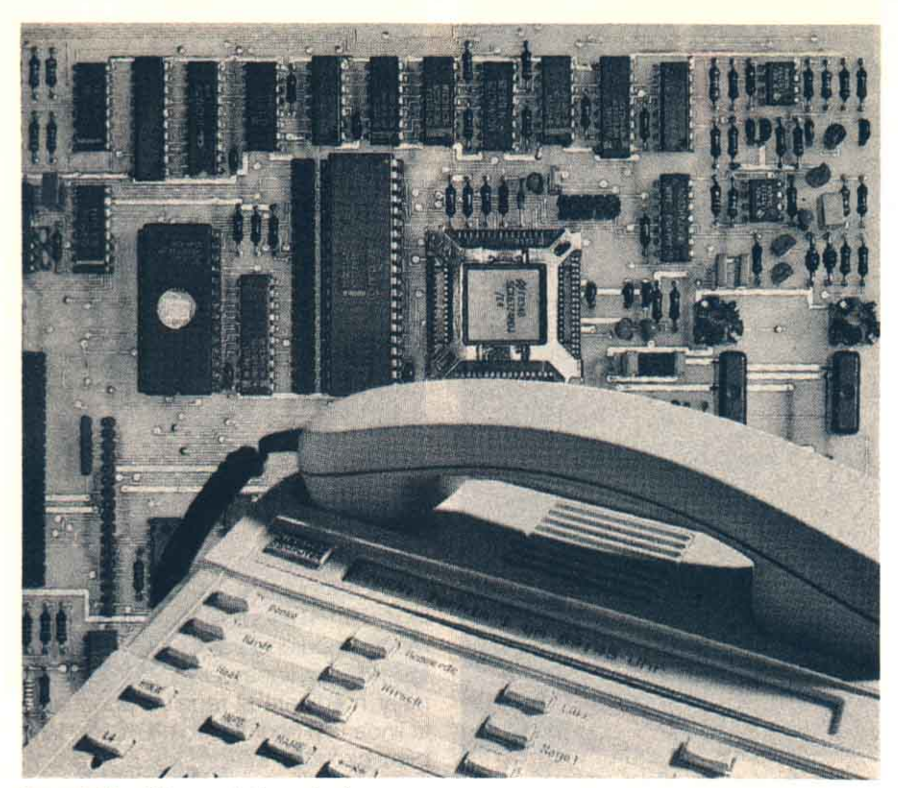


Foto 8: ***El teléfono y la tecnología, unidos por un mismo objetivo: enviar más información, más lejos en el menor tiempo.***

La velocidad de transmisión se mide en Baudios y hay que tenerla presente cuando se crea una red de transmisión.

La velocidad de transmisión, el número de bits a transmitir (5, 6, 7 u 8), el número de *bits* de parada (1, 1.5 ó 2) y la inclusión y detección de un *bit* de paridad, son algunos de estos parámetros. Este *bit* de paridad, se emplea para la detección de errores. Puede darse el caso de que la línea sea ruidosa y se falseen algunos *bits*. Si se desea la inserción de un *bit* de paridad, este se sitúa detrás de los bits de información y antes del *bit* de parada:

(Figura 8)

Si la paridad prefijada es par, es decir, si nosotros programamos paridad par en el registro correspondiente de la UART y el número de unos de la palabra a transmitir es par, se insertará un cero en el *bit* de paridad. De esta forma, si la línea telefónica ruidosa ha cambiado algún *bit*, la paridad no será la misma, esto es, el número de unos recibidos será diferente al no coincidir el *bit* de paridad recibido, con el de la palabra a transmitir y el de la paridad programada:

Número de bits -1 en el carácter	Paridad programada	Bits de paridad
Impar	Impar	0
Par	Impar	1
Impar	Par	1
Par	Par	0

Una secuencia normal del funcionamiento de las líneas de protocolo, se muestra en la figura 9. La activación de estas líneas se realiza en parte automáticamente por la UART.

(Figura 9)

Esta secuencia, corresponde a una conexión semiduplex.

En el instante a, el *interface* activa DTR, indicando que el equipo terminal está listo para transferir datos. En B, DSR, el *interface* está esperando a que el equipo de comunicaciones esté encendido. En C se activa RTS, indicando que el equipo terminal está listo para transmitir.

El *interface* espera entonces la activación de la línea CTS como respuesta. En D se activa CTS, transmitiendo el *interface* el dato. En E se desactiva RTS, indicando fin de la transmisión. Por último, en F la señal DSR vuelve al estado inicial, pudiéndo interpretarse como señal de desconexión.

A nivel de *software*, existe multitud de protocolos, independientes de las anteriores líneas de protocolo, en un principio.

Un protocolo bastante difundido es el ENQ/ACK (ENQuire ACKnowledge - preguntar/asentir). Cuando el equipo transmisor manda una línea de caracteres, le inserta al final de esta línea un carácter en ENQ (ASCII=05h). El transmisor espera un

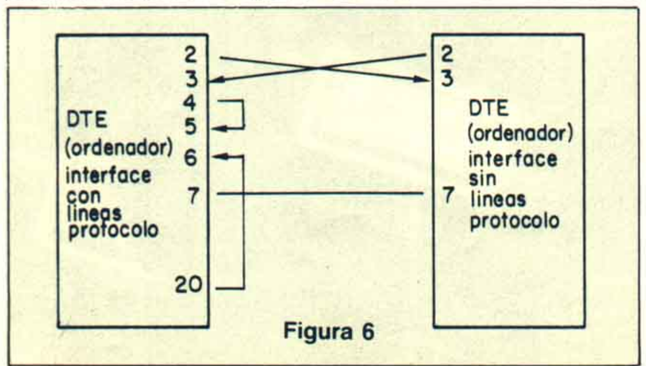


Figura 6

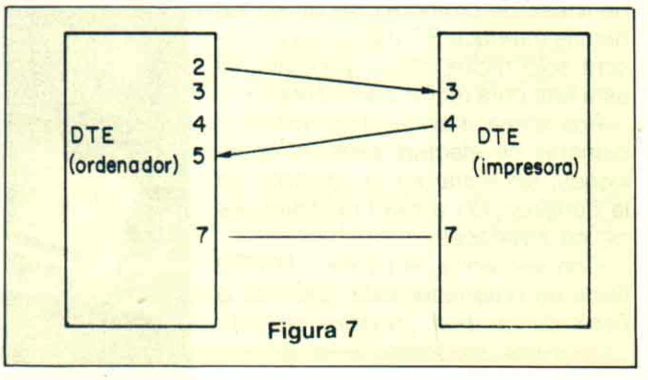


Figura 7

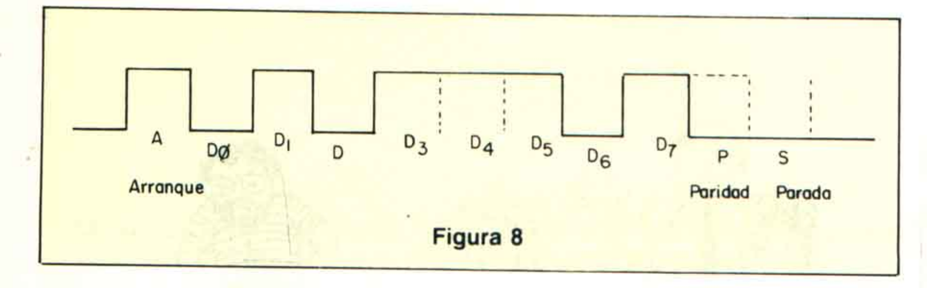


Figura 8

La pardidad es una forma de detectar los posibles errores que pudiera haber en una transmisión de datos.

carácter de ACK (ASCII=06h) del receptor antes de transmitir otra línea. El receptor responde con el carácter de ACK si la línea recibida no tiene errores y además dispone de suficiente espacio en su *butter* de recepción como para recibir otra línea de caracteres.

Otro protocolo bastante empleado es el XON/XOFF.

Durante la recepción de datos el equipo controla el estado de su *butter* de entrada. Cuando no hay suficiente espacio para albergar otro bloque de caracteres, el receptor manda un carácter de XOFF (ASCII 13h) al transmisor. Este deja de transmitir hasta que recibe un carácter XON (ASCII=11h) del receptor, indicándole que puede seguir transmitiendo.

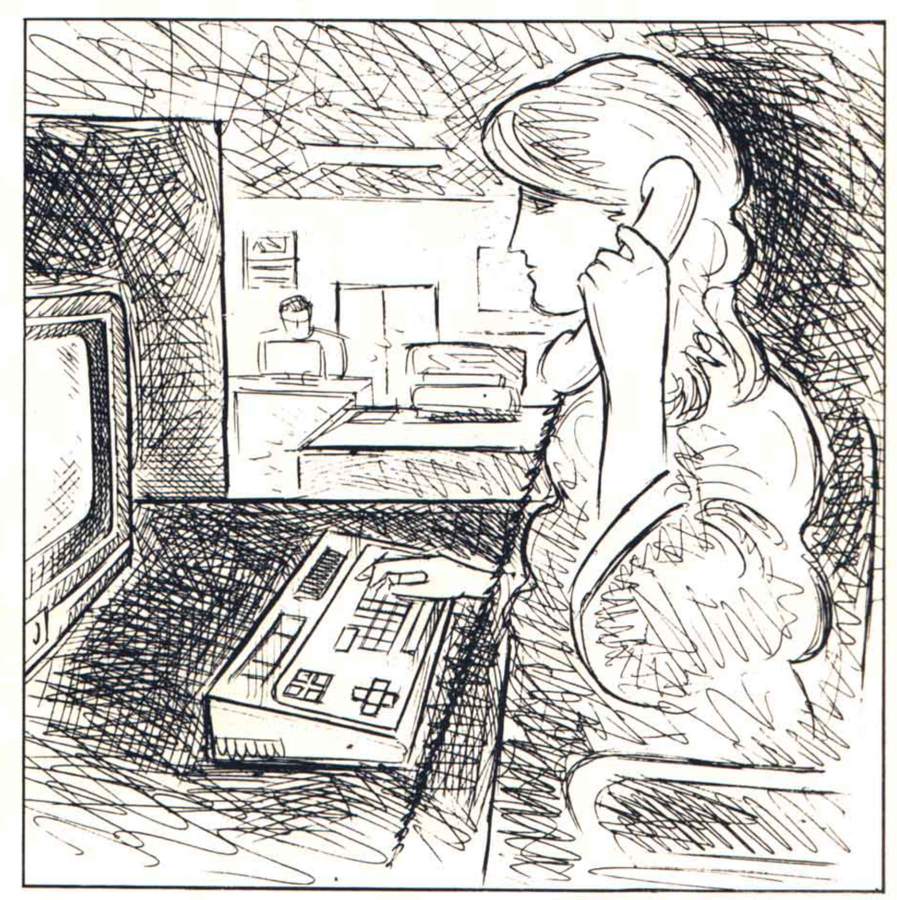
Como puede apreciarse, se trata de dos protocolos *software* elementales. Está claro que existen protocolo más complejos y sofisticados (HDLC/SDLC, CSMA, etc) que se emplean a otros niveles.

Es interesante señalar que la ISO ha presentado un modelo de comunicaciones de ordenadores que cuenta con 7 niveles.

La descripción del *interface* RS-232 C realizada aquí no pasaría del primer nivel, encontrándose sistemas de comunicaciones muy modernos que implementan hasta el cuarto nivel. Ello da una idea de la complejidad de los protocolos de comunicaciones, que tienen un componente de *software* muy elevado.

A nivel de ordenador personal hay que señalar la existencia de un programa de comunicaciones llamado KERMIT. Este programa es específico para cada modelo de ordenador pero tiene la particularidad de que una vez instalado permite la comunicación entre ordenadores de diferentes fabricantes.

Como punto final resaltar la actualidad del tema. Las comunicaciones de ordenadores, tanto a nivel de red local como a niveles de redes nacionales de conmutación de mensajes y conmutación de paquetes, haciendo resaltar que se trata fundamentalmente de programas de comunicaciones, apoyándose en un hardware específico en función del tipo de red a la que se quiera conectar.



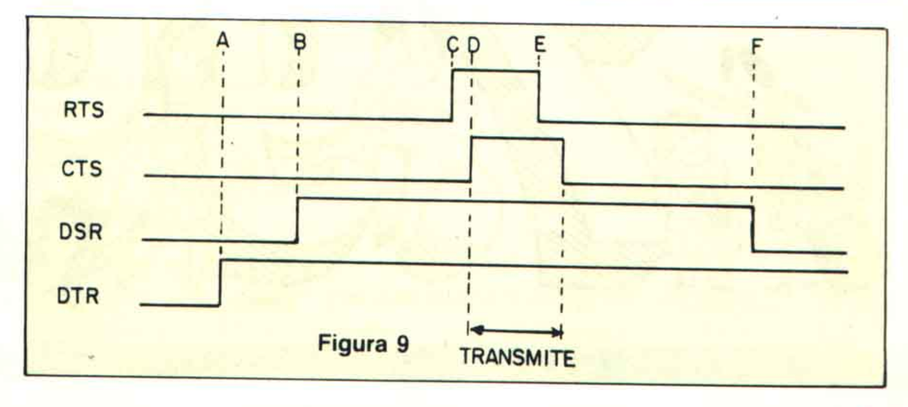
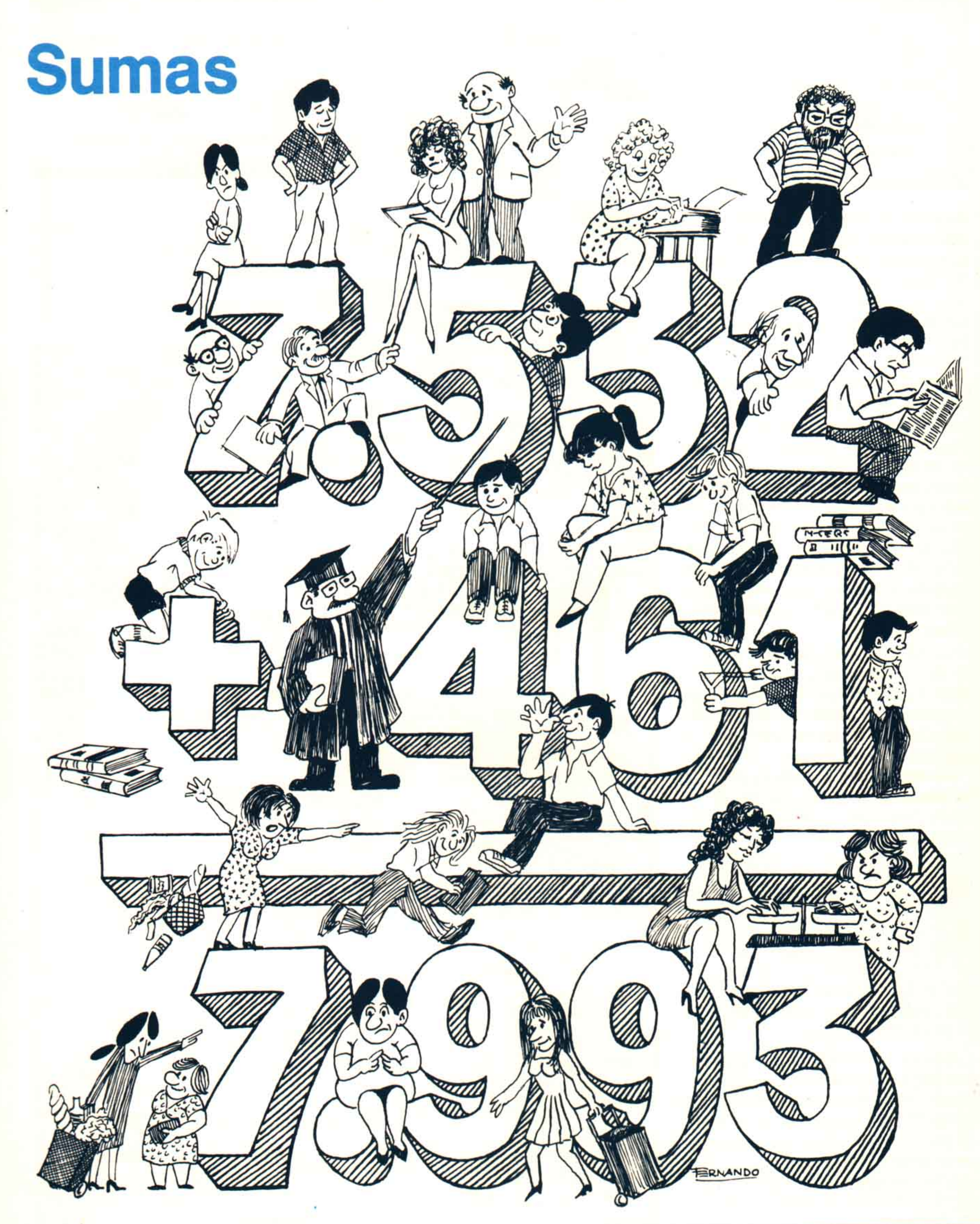


Figura 9

Sumas



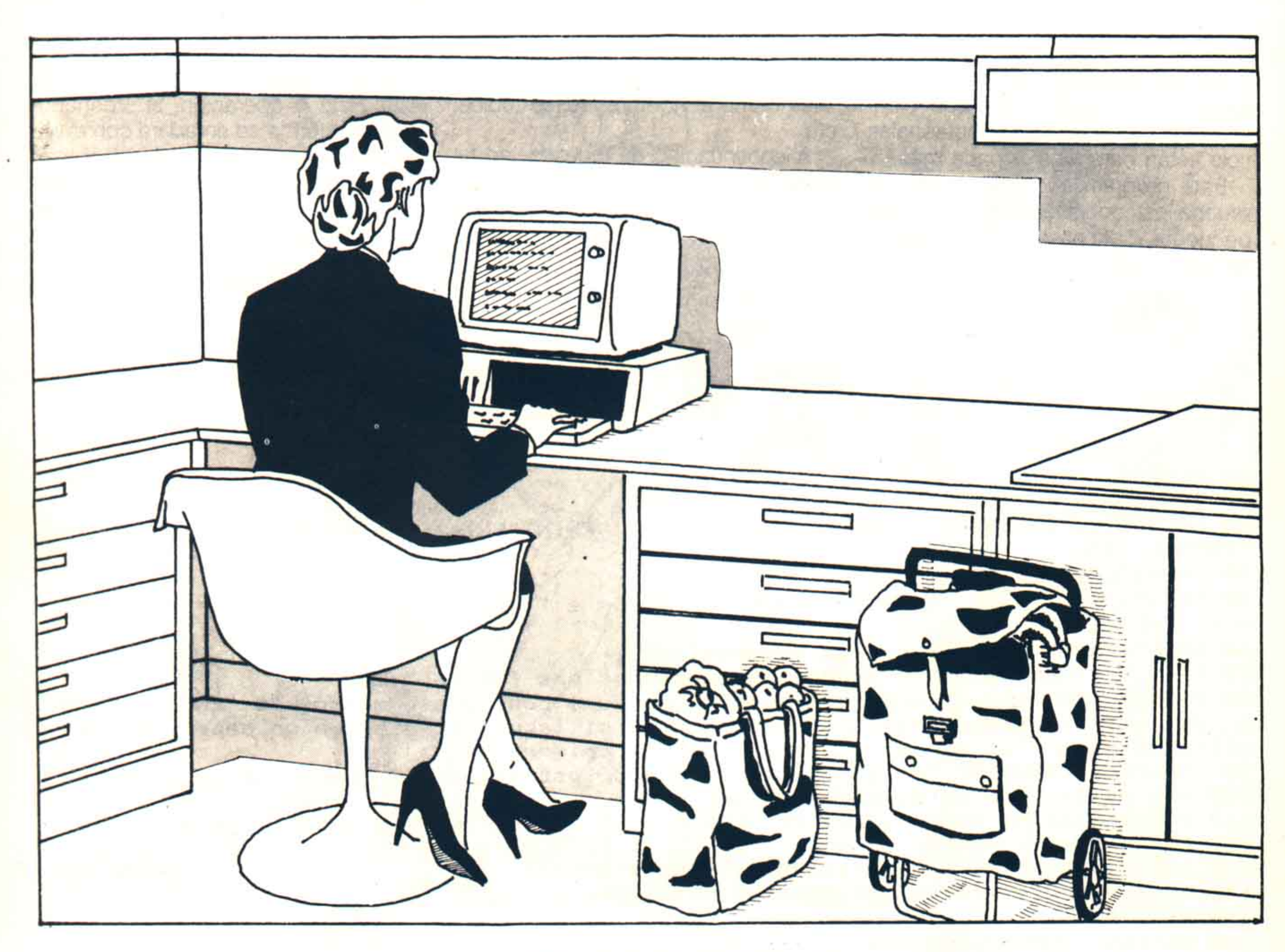
Es normal, hasta cierto punto, que todavía existan personas que piensen que los ordenadores personales sólo están para jugar y nada más.

Este programa no sólo, pondrá a prueba los conocimientos de los más pequeños de la casa, sino también el de los mayores. Se empienza respondiendo a una serie de preguntas tales como el nombre, sus estudios, etc.

A continuación se muestra, en tamaño gigante, las dos cifras a sumar. Este efecto se consigue gracias al comando SCREEN 3. La operación se realiza de la misma manera que si lo hiciéramos a mano. Una vez finalizada la operación, el ordenador nos felicitará o se enfadará con el estudiante, según el resultado obtenido. Ideal para los pequeños, esperamos que disfruten y hagan del estudio una diversión, que es como mejor se aprende.

```
10 CLS:KEY OFF
20 REM sumas
30 LOCATE 0,10: PRINT" SUMAS ":LOCATE 0,20: PRINT"Pulsa una tecla para continuar
":GOSUB 340
40 INPUT"Como te llamas";N$
50 PRINT:PRINT"bien, ";N$;" quieres aprender a ":INPUT"sumar";S$:IF S$="n" OR S$
="N" THEN END
60 PRINT:INPUT"cuantas cifras quieres";NC
70 IF NC>3 THEN PRINT"Tu necesitas emociones mas fuertes":END
80 PRINT:PRINT"Ahora te va a aparecer una suma con  dos sumandos de ";NC;" cifra
s. Vas a ir introduciendo las  cifras como si lasescribieses en un papel. Si la
ultimaes mayor de 10,  solo debes poner la primera cifra"
90 LOCATE 0,20:PRINT "Pulsa una tecla cuando estes listo": GOSUB 340
100 REM definicion de sumandos
110 P=10^NC:A=INT(RND(-TIME)* P ):FOR N=1 TO 20:NEXT N:B=INT(RND(-TIME)* P )
120 A$="":B$="": FORN=1 TO NC
130 A=INT(RND(-TIME)*10):B=INT(RND(-TIME)*10)
140 A$=A$+RIGHT$(STR$(A),1):B$=B$+RIGHT$(STR$(B),1)
150 NEXT N
160 SCREEN 3:OPEN"grp:"AS#1
170 PRESET (100,20):PRINT#1, A$:PRESET(100,60):PRINT#1, B$
180 LINE(80,100)-STEP(100,0),1
190 FOR S=NC TO 1 STEP-1
200 B= VAL((MID$(A$,S,1))):C=VAL(MID$(B$,S,1)):SUM=B+C+CAR
210 IF SUM>9 THEN SUM=SUM-10:CAR=1 ELSE CAR=0
220 I$=INKEY$
230 IF I$="" THEN GOTO 220
240 A=VAL(I$)
250 IF A=SUM THEN GOTO 270
260 GOSUB 350:PRINT#1, "mal":GOTO 220
270 GOSUB 350: PRINT#1, "bien"
280 PS=40+S*33:PRESET (PS,110):PRINT#1,SUM
290 IF S=1 AND CAR=1 THEN PRESET (42,110):PRINT#1,1
295 IF S=1 THEN PRESET(80,145):PRINT#1,"      !"
300 NEXT S:GOSUB 340:CLOSE#1
310 SCREEN 0: PRINT "Bien, lo has conseguido. ":INPUT "quieres hacer otra";S$
320 IF S$="s" OR S$="S" THEN RUN 60
330 CLS:KEY ON:END
340 IF INKEY$="" THEN GOTO 340 ELSE CLS:RETURN
350 PRESET (80,145),4:COLOR 4:PRINT #1,"█████":COLOR15:PSET(80,145):RETURN
```

Contabilidad Casera



Los programas de aplicación para ordenadores MSX, no abundan demasiado y menos cuando nos lo envía un lector. Nuestro amigo Miguel Maiquez Balibrea de Murcia, nos ha enviado esta aplicación, que además de ser sumamente interesante, es una utilidad en todo el sentido de la palabra.

Al ejecutar el programa, nos encontramos con un menú de 8 opciones. Estas son las siguientes:

1) Dar alta cuenta. En ella se tiene acceso a la apertura de hasta 20 cuentas que son las que nos van a servir para controlar los gastos e ingresos.

2) Introducir asientos. Podrá introducir hasta un total de 200 apuntes, indicando a la vez la cantidad y si ésta se coloca al Debe o Haber de la cuenta.

3) Listado general del Diario. Se nos indicará los movimientos que se han efectuado con el formato, fecha, concepto, código de cuenta, contrapartida e importe. Los valores que se sitúan al Haber se les pone un signo negativo.

4) Listado general de Altas. Listará todas las altas habidas hasta el momento y en donde se indica el número del código y su título.

5) Listado por cuentas. Util para comprobar el movimiento de cada una de las cuentas abiertas.

6) Balance. Lógicamente, obtendremos el balance de todas las cuentas abiertas y que han tenido movimiento, indicando con el signo (–) las correspondientes al Haber.

7) Grabación de ficheros. Para grabar todos los movimientos.

8) Carga de ficheros. En donde se cargarán los movimientos de sesiones anteriores.

Miguel Maiquez Balibrea (Murcia)

```
1 REM*******************************
2 REM CONTABILIDAD-MANUEL MAIQUEZ
3 REM*******************************
5 CLS
6 CLEAR 6000
7 X$="                                        "
10 DIM A(200),A$(200),B$(200),C$(200),B(200),D$(20)
15 CLS
16 KEY OFF
20 PRINT "    M  E  N  U  "
30 PRINT"   ---------------"
40 PRINT:PRINT:PRINT
50 PRINT" 1- DAR ALTA CUENTA
60 PRINT" 2- INTRODUCIR ASIENTOS
70 PRINT" 3- LISTADO GENERAL DIARIO
80 PRINT" 4- LISTADO GENERAL DE ALTAS
90 PRINT" 5- LISTADO POR CUENTAS
95 PRINT" 6- BALANCE
96 PRINT" 7- GRABACION DE FICHEROS
97 PRINT" 8- CARGA DE FICHEROS
100 PRINT:PRINT:PRINT
110 INPUT"QUE OPCION DESEA";D
120 IF D<1 OR D>8 THEN 15
130 ON D GOTO 150,330,600,900,1000,1200,1400,1600
150 REM DAR DE ALTA UNA CUENTA
160 CLS
170 PRINT"    A  L  T  A  S  "
180 PRINT"===================="
190 PRINT:PRINT:PRINT;
195 PRINTTAB(17)"CUENTAS LIBRES"19-L1
200 L1=L1+1
205 IF L1>20 THEN PRINT"FICHERO DE CUENTAS COMPLETO":GOTO 15
210 PRINT" CUENTA NUMERO" L1
220 PRINT
230 PRINT" TITULO:    .........."
240 LOCATE 10,7:INPUT A1$
250 IF LEN(A1$)>9 THEN LOCATE0,7:PRINTX$:LOCATE 0,7: GOTO 230
260 D$(L1)=A1$
270 PRINT:PRINT:PRINT:PRINT
280 PRINT"DESEA DAR ALGUN ALTA MAS"
290 INPUT "S / N"; F$
300 IF F$="S"THEN 160
310 IF F$="N"THEN CLS:GOTO 20
320 IF F$<>"S" OR F$<>"N" THEN 290
325 REM INTRODUCCION DE ASIENTOS
330 CLS
340 L=L+1
345 IF L>200 THEN 15
360 PRINT"INTRODUCCION DE ASIENTOS"
365 PRINT"========================"
370 PRINT:PRINT
371 PRINTTAB(17)"APUNTE NUM " L
372 PRINT
380 INPUT"CODIGO DE CUENTA";I
385 IF I>20 THEN CLS:GOTO 360
390 PRINT:PRINT D$(I)
400 PRINT:PRINT:A(L)=I:A$(L)=D$(I)
410 PRINT"FECHA  DDMMAA"
```

```
420 LOCATE 5,11:INPUT B$(L)
425 IF LEN (B$(L))>6 THEN LOCATE 0,11:PRINTX$:LOCATE 0,11:GOTO 410
430 PRINT
440 PRINT"CONCEPTO  ........"
450 LOCATE 8,13:INPUT C$(L)
460 IF LEN(C$(L))>8 THEN LOCATE 0,13:PRINTX$:LOCATE0,13:GOTO440
465 PRINT
470 INPUT"IMPORTE";B(L)
475 PRINT
480 INPUT" D / H "; B1$
490 IF B1$="H" THEN B(L)=B(L)*-1
495 IF B1$<>"D" ANDB1$<>"H"THEN480
496 PRINT:PRINT TAB(15)" LIBRES "200-L"APUNTES
500 INPUT"ASIENTO CORRECTO S / N";A2$
525 IF A2$="S"  THEN 530
526 IF A2$="N" THEN CLS:GOTO 360
527 IFA2$<>"S" OR A2$<>"N"THEN 500
528 PRINT
530 INPUT"DESEA  ALGUNA ENTRADA MAS S / N";A2$
535 H=0
540 IF A2$="S" THEN 330 ELSE15
600 REM LISTADO GENERAL DIARIO
610 CLS
630 PRINT"FECHA  CONCEP  COD  TITULO     CANTIDAD"
640 PRINT"======================================="
650 FOR I=1 TO L
660 PRINTB$(I);:PRINT" ";: PRINTUSING"\       \";C$(I);:PRINT" ";: PRINTUSING"##"
;A(I);:PRINT" ";: PRINTUSING"\         \";A$(I);:PRINT" ";:PRINTUSING"##,###,###";
B(I)
670 NEXT I
675 LOCATE0,23
680 INPUT"PARA MENU PULSE 0";A3
690 IF A3=0 THEN 15
695 IF A3<>0 THEN 680
900 REM LISTADO CODIGOS CUENTAS
910 CLS
928 PRINT"CODIGO  TITULO CUENTA"
929 PRINT"====================
930 PRINT
940 FOR I=1 TO L1
950 PRINTUSING"##";I;
960 PRINT" ";
970 PRINTD$(I)
980 NEXT I
985 LOCATE0,23
990 INPUT"PARA MENU PULSE 0";A3
991 IF A3=0 THEN 15
1000 REM LISTADO POR CUENTAS
1010 CLS
1020 INPUT"CUENTA A LISTAR";J
1030 CLS
1040 PRINT"CUENTA NUMERO" J
1050 PRINT D$(J)
1060 PRINT"======================================="
1070 PRINT"FECHA    CONCEPTO        IMPORTE  "
1080 PRINT"======================================="
1090 FOR I=1 TO L
```

```
1100 IF A(I)=J THEN PRINT B$(I);:PRINT" ";:PRINTC$(I);:PRINT"  ";:PRINTUSING"
##,###,###";B(I):K=K+B(I)
1110 NEXT I
1120 PRINT USING"                              ##,###,###";K:K=0
1130 LOCATE 0,23
1140 INPUT"PARA MENU PULSE 0 NUEVO LISTADO 1";A2
1150 IF A2=0 THEN 15 ELSE 1010
1200 REM BALANCE
1210 CLS
1215 PRINT" B A L A N C E   (+ DEBE - HABER )
1216 PRINT"==============
1220 L2=0
1230 L2=L2+1
1235 IF L2=21 THEN 1305
1240 FOR I=1 TOL
1250 IF A(I)=L2 THEN M=M+B(I)
1260 NEXT I
1270 IF M=0 THEN 1230
1275 M1=M1+M
1280 PRINTUSING"\          \";D$(L2);
1290 PRINTUSING" ##,###,###";M:M=0
1300 GOTO 1230
1305 PRINT
1306 PRINTUSING"               ##,###,###";M1
1307 M1=0
1310 LOCATE0,23:INPUT"PARA MENU PULSE 0";A2:L2=0:IF A2=0 THEN 15
1400 REM RUTINA DE GRABACION
1410 CLS
1420 PRINT"PONGA EL CASETTE PARA GRABAR Y CUANTO  ESTE DISPUESTO PUSE 0":INPUT A
2
1430 CLS:PRINT"GRABANDO"
1440 OPEN "CAS:CONT"FOR OUTPUT AS 1
1450 PRINT#1,L:PRINT#1,L1
1460 FOR I=1 TOL
1470 PRINT#1,A(I):PRINT#1,A$(I):PRINT#1,B$(I):PRINT#1,C$(I):PRINT#1,B(I)
1480 NEXT I
1490 FOR I=1 TO 20
1500 PRINT#1,D$(I)
1510 NEXT I
1520 CLOSE 1
1530 CLS
1540 PRINT"TODO GRABADO PUEDE APAGAR"
1550 PRINT"PROGRAMA TERMINADO":END
1600 REM CARGA FICHEROS
1610 CLS
1620 PRINT"PONGA EL CASETTE PARA QUE DE INFORMACI-ON Y CUANTO ESTE DISPUESTO PUL
SE 0":INPUT A2
1630 CLS:PRINT"CARGANDO EL FICHERO
1640 OPEN "CAS:CONT"FOR INPUT AS 1
1650 INPUT#1,L:INPUT#1,L1
1660 FOR I=1 TO L
1670 INPUT#1,A(I):INPUT#1,A$(I):INPUT#1,B$(I):INPUT#1,C$(I):INPUT#1,B(I)
1680 NEXT I
1690 FOR I=1 TO 20
1700 INPUT#1,D$(I)
1710 NEXT I
1720 CLOSE 1
1730 GOTO 15
```



Carrera de caballos

Ahora que finalizado la temporada futbolística y, con ella, la de hacer quinielas (se puede hacer, pero tienen poco interés), la inmensa mayoría de los aficionados se vuelca hacia los caballos. Encuentran una válvula de escape, animando y apostando por su caballo favorito.

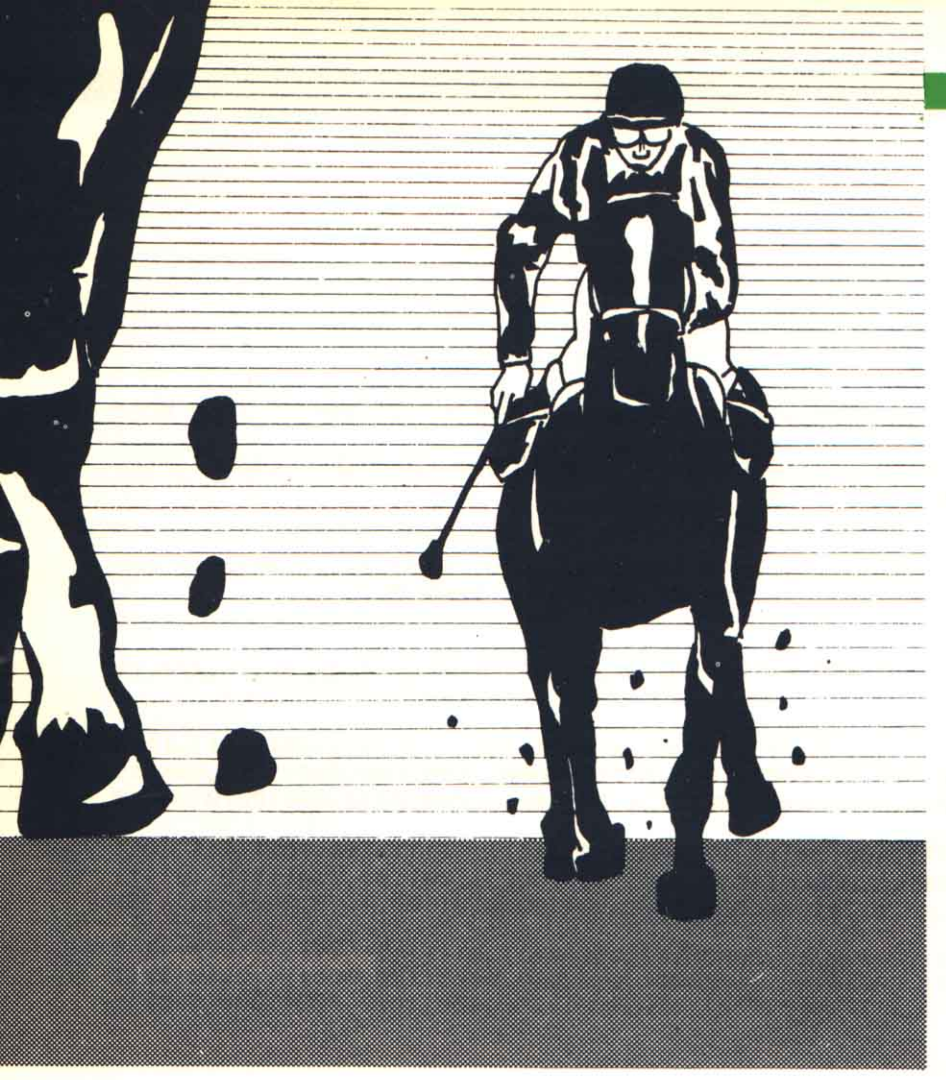
Actualmente, la quiniela hípica se está poniendo de moda; tanto es así, que ha llegado a suplir a la quiniela futbolística (que se lo pregunten a la Federación cuando ocurrió la huelga de futbolistas). En este sentido, los aficionados la han convertido en la segunda opción.

Les presentamos un programa que les hará sentirse como si estuvieran en el hipódromo, con todos los alicientes que poseen.

Las carreras tienen lugar en el hipódromo de Pastizales, donde, como indican las instruciones, hay un completo programa de carreras en las que podremos conseguir importantes sumas de dinero. Claro está que para ello tendremos que apostar por el caballo adecuado, puesto que todos no corren por igual y, además, podemos encontrarnos, con que nuestro favorito no tiene su día, haciéndonos perder el dinero y la confianza que en él teníamos.

Al ejecutar el programa, se nos presentará unas instruciones a seguir, a continuación de las cuales se nos

```
10 REM     CABALLOS
20 REM     INSTRUCCIONES
30 KEY OFF: WIDTH 37: CLS: LOCATE 10,0:  PRINT "CABALLOS"
40  PRINT : PRINT"Espera un ratito, estoy algo ocupado":GOSUB 470:LOCATE 0,1
50  PRINT : PRINT " Hola, bienvenido al gran  hipodromo de Pastizales.": PRINT "
Tenemos hoy un completo programa de  4 fabulosas carreras, todas ellas muyemocio
nantes"
60  PRINT : INPUT "Antes de nada, sin animo de cotilleo,Cuanto dinero has traido
 ";DIN
70 IF DIN<=0 THEN  PRINT "Lo siento, pero deberas volver otro  dia con tu tia ri
ca. Adios !": END
80 PRINT :  PRINT "No esta mal. Tienes la oportunidad deconseguir mucho mas apos
tando en cadacarrera.": INPUT "Te atreves";Q$
90 IF Q$="n" OR Q$="N" THEN  PRINT "Tu verás. Hasta otra": END
100 DATA "Silver devil","Bonobús gastado","herradura rota"
110 DATA "Monedero falso","Gata loca","Simplón"
120 DATA "Barriga floja","Flower boy","rata verde"
130 DATA "Cabra saltarina","Bombilla fundida","Crazy horse"
140 DIM A$(12)
150 RESTORE 100
160 FOR N=1 TO 12
170 READ A$(N):NEXT N
180 FOR C%=1 TO 4
```



preguntará el dinero que poseemos, al igual que la cantidad a apostar.

Una vez hecha la apuesta pasamos a la acción. Desde nuestro privilegiado puesto de observación, podremos contemplar el desarrollo de la carrera, además de tener el honor de dar la salida.

Al final de la carrera, el ordenador, que para eso está nos indicará la cantidad de dinero ganado o perdido, y también el balance en cada momento.

Es un buen programa que nos hará pasar un rato divertido, a la vez que nos permite apostar con los amigos sin salir de casa. Suerte, y que gane el mejor.

```
190 FOR N=1 TO 3
200 BET(N)=INT(RND(-TIME)*5)+2
210 IN= INT(RND(-TIME)*12)+1
220 IF N=1 THEN GOTO 250
230 FOR L=N-1 TO 1 STEP -1: IF B(L)=IN THEN GOTO 210
240 NEXT L
250 B(N)=IN
260 NEXT N
270 REM   APUESTAS
280 CLS
290 LOCATE 0,3: PRINT "Muy bien, en la carrera num. ";C%;"van a  correr los sigu
ientes caballos con diferentes apuestas a su favor:"
300 FOR N=1 TO 3: PRINT N;"     "; A$(B(N));TAB(15)BET(N):NEXT N
310  PRINT : INPUT "Por cual te decides";CAB:INPUT"Cuanto apuestas por el";AP
320 IF AP> DIN THEN  PRINT "Perdona pero no tienes tanto. Intentalo otra vez de
nuevo": LOCATE 0,9: GOTO 310
330  PRINT  "Atento, que empezamos
       "
340 GOSUB 590
350 IF GAN=CAB THEN GOTO 360 ELSE 400
360 SCREEN 0:  PRINT "Estupendo, has ganado": PRINT "Vamos a ver cuanto debes re
coger.": PRINT AP*BET(CAB);" pts"
370  PRINT "que si lo sumamos a lo que tenias da:": DIN=DIN+(AP*BET(CAB))
```

```
380  PRINT DIN;" pts"
390  PRINT : INPUT "Te atreves a seguir (s/n)";Q$: IF Q$="s" OR Q$="S" THEN GOTO
 450 ELSE GOTO 440
400 SCREEN 0:  PRINT "Lo siento, tu lo has visto igual que yo. Debo descontarte
el dinero de la apuesta. Te quedan :";
410 DIN=DIN-AP
420 PRINT DIN
430 IF DIN>0AND C<4 THEN PRINT"No obstante te queda algo. Podrias intentarlo otr
a vez.":INPUT"Te atreves (s/n)";Q$: IF Q$="s" OR Q$="S" THEN GOTO 450
440  PRINT "Adios": END
450 NEXT C%
460  PRINT " Estupendo, has conseguido ";DIN;"pts": PRINT "Felicidades, hasta ot
ra":END
470 REM Carrera
480 DATA 0,0,00000001,00000011,00000111,00000111,01111111,1111111,10111111,1011
1111,00110100,01100100,11001000,10000000,0,0,0,11000000,11001100,00011110,001111
11,01111011,11111001,11111000,11111000,00011110,00010010,00010010,00010010,00100
000,0,0
490 DATA 0,0,00000011,00000111,00000111,00000011,01111111,1111111,10111111,1011
1111,00110000,00011000,00001100,00000110,0,0,0,11000000,11001100,00011110,101111
11,01111011,11111001,11111000,11111000,00011000,00110000,01100000,01000000,11000
000,0,0
500 DATA 0,00000011,00000011,00000111,00000111,00000111,01111111,1111111,1011111
1,10111111,10111000,00011000,00011000,00010000,0,0,0,0,00001100,00011110,1011111
1,01111011,11111001,11111000,11111000,00011000,00011000,00011000,00011000,000010
00,0,0
510 REM def sprites
520 S$="":T$="":R$=""
530 RESTORE 470:FOR N=1 TO 96
540 READ A$: A$=CHR$(VAL("&b"+A$))
550 IF N<33 THEN S$=S$+A$
560 IF N>33 AND N<65 THEN R$=R$+A$
570 IF N>65 THEN T$=T$+A$
580 NEXT N : RETURN
590 SCREEN 2,2:COLOR 1,11,11
600 SPRITE$(1)=T$
610 SPRITE$(2)=T$
620 SPRITE$(3)=T$
630 LINE(0,50)-(255,125),3,BF
640 LINE (0,125)-(25,50),15: LINE(215,125)-STEP(20,-75),15
650 OPEN"grp:"AS#1:PRESET(233,45):  PRINT #1,"∞"
660 SPRITE$(1)=T$
670  PRESET (0,0):PRINT#1,"  Tu das la salida pulsando          una tecla"
680 DATA 15,60,10,80,5,100
690 RESTORE 680: FOR N=1 TO 3:FOR L=1 TO 2: READ P(N,L):NEXT L:NEXT N
700 FOR N=1 TO 3: PUT SPRITE(N),(P(N,1),P(N,2)),(15-N):NEXT N
710 IF INKEY$="" THEN 710
720 FOR S=1 TO 2
730 FOR N=1 TO 3
740 PUT SPRITE(N),(P(N,1),P(N,2)),(15-N)
750 P(N,1)=P(N,1)+INT(RND(-TIME)*4)+1
760 IF S=1 THEN SPRITE$(N)=R$
770 IF S=2 THEN SPRITE$(N)=S$
780 IF P(N,1)>(210+N*5) THEN GOTO 810
790 NEXT N
800 NEXT S: GOTO 720
810 GAN=N: PRESET (10,200): PRINT#1," EL GANADOR HA SIDO ";GAN: CLOSE#1: RETURN
```

Circulos

Creo que todos los lectores estarán más que acostumbrados por las capacidades gráficas que ofrece el MSX. Aunque no las conozcan a fondo, es bueno estimular la curiosidad y abrir las puertas para que otros mejoren lo que unos han conseguido.

Este es un caso de estos. El programa es simple, pero representativo, ya que a base de combinar el comando CIRCLE con bucles FOR--NEXT–STEP, se logra un efecto interesante y curioso. Huelga decir que el programa está abierto a todas las mejoras necesarias, así lo indica Fernando Brias Cobos, autor de este pequeño programa y para él nuestra recompensa.

Fernando Brias Cobos (Granada)

```
10 '*******************************
20 '************* msx *************
30 '***** Graficos / Dibujos *******
40 '**** Fernando Brias Cobos ******
50 '*******************************
60 SCREEN 3
70 OPEN"GRP:"AS#1
80 PRINT #1,"Gráficos"
90 PRINT #1," por"
100 PRINT#1,"":PRINT #1," F.B.C."
110 FOR X=0 TO 2000:NEXT X
120 SCREEN2:COLOR15,4,4
130 FOR X%=0 TO 200 STEP 2
140 CIRCLE(X%,X%),5,15
150 CIRCLE(256-X%,192-X%),5,15
160 CIRCLE(256-X%,X%),5,15
170 CIRCLE(X%,192-X%),5,15
180 NEXT X%
190 FOR X%=200 TO 50 STEP -2
200 CIRCLE(X%,X%),5,4
210 CIRCLE(256-X%,192-X%),5,4
220 CIRCLE(256-X%,X%),5,4
230 CIRCLE(X%,192-X%),5,4
240 NEXT X%
250 FOR X%=0 TO 157 STEP 2
260 CIRCLE(50+X%,50),5,3
270 CIRCLE(206-X%,142),5,3
280 NEXT X%
290 FOR X%=0 TO 90 STEP 2
300 CIRCLE(206,50+X%),5,9
310 CIRCLE(50,142-X%),5,9
320 NEXT X%
330 FOR X%=1 TO 15 STEP 2
340 FOR S%=0 TO 60 STEP 2:CIRCLE(127,95),S%,X%,,,2.4
350 NEXT S%,X%
360 FOR X%=0 TO 70 STEP 2
370 CIRCLE(X%,X%),X%,10
380 CIRCLE(256-X%,X%),X%,2
390 CIRCLE(256-X%,192-X%),X%,13
400 CIRCLE(X%,192-X%),X%,8
410 NEXT X%
420 SCREEN 2:COLOR 15,1,1
430 CLOSE:OPEN"grp:"AS#1
440 PRESET(5,5):PRINT#1,"Estas son algunas posibilidades"
450 PRESET(40,20):PRINT#1,"del G.M.L del msx"
460 GOTO 130
```



Reinas de ajedrez

Ahora es tiempo de vacaciones, muchos lectores se abandonarán al placer de entretenerse con juegos tales como el ajedrez, damas y demás juegos de mesa. La mayoría habrá jugado al ajedrez esas tardes bochornosas, cuando salir de casa se convertía más en hazaña que en otra cosa.

El presente programa plantea un problema de no fácil solución, aunque éstas existen, de hecho hay un total de 60 soluciones al problema.

La cuestión está en colocar ocho reinas en un tablera de ajedrez sin que se coman unas a otras. Las soluciones no son difíciles, pero hay que pensar en ellas, a cada cual más complicada. Prueba de ellos son las puntuaciones que irá dando el ordenador por reina colocada.

El juego es un pasatiempo muy entretenido, que permitirá desarrollar la capacidad de memoria y la velocidad de razonamiento. Un buen juego sería a ver quién encuentra una solución al problema en el menos tiempo posible.

Juan Manuel Camaño (Madrid)

```
10 ' >>> JUEGO DE LAS OCHO REINAS <<<
20 '
30 '
40 DIM A(10,10)
50 SCREEN0:CLS:KEYOFF
60 COLOR 15,4,4
70 ON STOP GOSUB 1970
80 STOP ON
90 ON ERROR GOTO 1970
100 '
110 ' >>>>> INSTRUCIONES <<<<<
120 '
130 LOCATE 14,1,0:PRINT"<<HOLA>>
140 LOCATE0,5,0:PRINT"SOY UN JUEGO MUY SENCILLO"
150 PRINT"SOLO DEBES COLOCAR >8< REINAS"
160 PRINT"EN EL TABLERO DE AJEDREZ"
170 PRINT"PERO ATENCION":PRINT
180 PRINT:PRINT">> NO PUEDEN COMERSE UNAS A OTRAS <<"
190 PRINT:PRINT
200 PRINT"1 - PON EL VALOR HORIZONTAL"
210 PRINT
220 PRINT"2 - PON EL VALOR VERTICAL"
230 PRINT:PRINT"SENCILLO VERDAD?"
240 PRINT:PRINT"PULSA [0] Y [RETURN] PARA EMPEZAR"
250 INPUT A$:IFA$=CHR$(48)THEN 280 ELSE CLS
260 GOTO 240
270 '
280 ' >>> MUSICA DE PRESENTACION <<<
290 '
300 V$="M8000T120S204G903A9R1003L9AA+04CAR1904FL504FR49L9C03AR1003AR30A+03A04L5C
R30C03A+"
310 W$="M8000T120S204G903A9R1003L9AA+04CAR1904FL504FR49L9C03AR1003AR30A+03A04L5C
R30C03A+"
320 '
330 '>>>> FIGURA PRESENTACION <<<<<
340 '
350 CLS:COLOR 0,0,0:SCREEN 2
360 A$= "S=I;C=R;BURD2L2U2RBD"
370 B$= "S=J;C=B;BURD2L2U2RBD"
380 R=8:B=15
390 DRAW "BM200,100"
400 J=0
410 FOR K=0 TO 7
420 FOR I=4 TO 240 STEP 18
430 IF I>78 THEN J=I-72
440 DRAW"XA$;"
450 DRAW"XB$;"
460 NEXT I
470 B=B-2:R=5+1
480 NEXT K
490 PLAY V$,W$
500     COLOR 8,0,0
510 SCREEN 3
520 OPEN"GRP:"FOR OUTPUT AS #1
530 PSET (80,24)
540 PRINT#1,"LAS"
550 PSET(64,88)
560 PRINT#1,"OCHO"
570 PSET (32,152)
580 PRINT#1,"REINAS"
590 FOR I=0 TO 1200:NEXT I
```

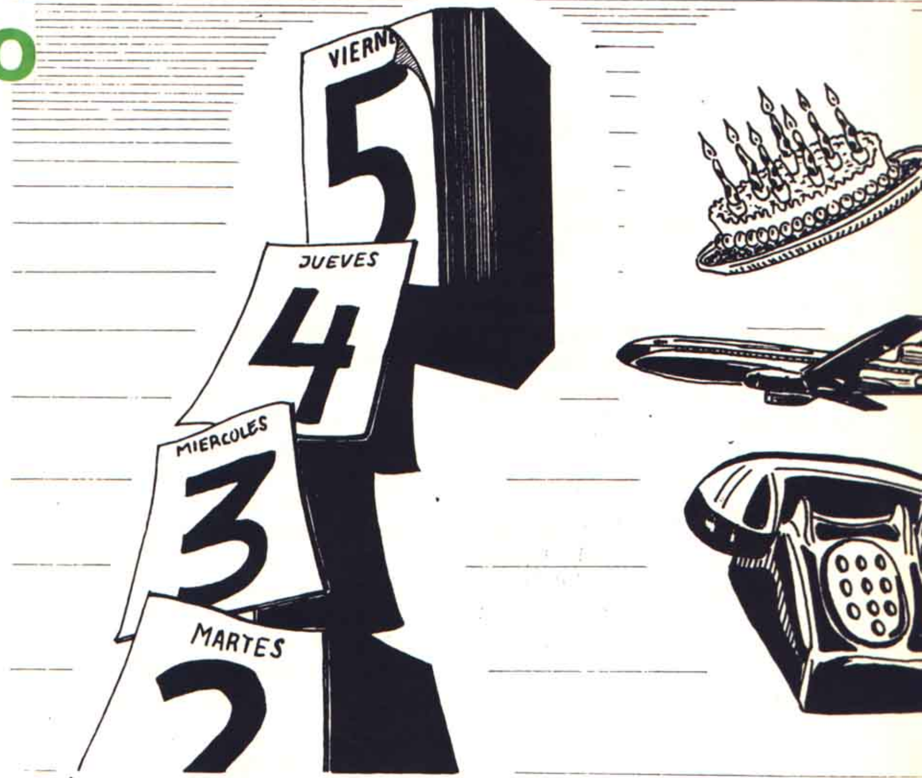
```
600 GOTO 770
610 '
620 '>>>> DEFINICION DE SPRITES <<<<
630 '
640 A$=CHR$(&H5)+CHR$(&H3)+CHR$(&H6)+CHR$(&HE)+CHR$(&H19)+CHR$(&H21)+CHR$(&H27)+
CHR$(&HB)
650 B$=CHR$(&H13)+CHR$(&H23)+CHR$(&H7)+CHR$(&H7)+CHR$(&HF)+CHR$(&H1F)+CHR$(&H3F)
+CHR$(&H7F)
660 C$=CHR$(&H40)+CHR$(&H80)+CHR$(&H8)+CHR$(&H8)+CHR$(&H94)+CHR$(&H8)+CHR$(&HC8)
+CHR$(&HA8)
670 D$=CHR$(&HA8)+CHR$(&H98)+CHR$(&H88)+CHR$(&H88)+CHR$(&HC8)+CHR$(&HC8)+CHR$(&H
C8)+CHR$(&HE8)
680 SPRITE$(0)=A$+B$+C$+D$
690 SPRITE$(1)=A$+B$+C$+D$
700 SPRITE$(2)=A$+B$+C$+D$
710 SPRITE$(3)=A$+B$+C$+D$
720 SPRITE$(4)=A$+B$+C$+D$
730 SPRITE$(5)=A$+B$+C$+D$
740 SPRITE$(6)=A$+B$+C$+D$
750 SPRITE$(7)=A$+B$+C$+D$
760 RETURN
770 '
780 '>>> PANTALLA <<<
790 '
800 SCREEN 1,2
810 COLOR 7 ,0,0
820 FORX=1TO16STEP4:FORY=1TO16STEP4
830 GOSUB 1070:NEXTY:NEXTX
840 FORX=3TO16STEP4:FORY=3TO16STEP4
850 GOSUB1070:NEXTY:NEXTX
860 KEYOFF:
870 GOSUB 1080
880 R=0:S=0
890 FORY=1TO16STEP2:S=S+1
900 LOCATE18,Y,0:PRINTS:NEXTY
910 FORX=0TO15STEP2:R=R+1
920 LOCATEX,18,0:PRINT R:NEXTX
930 GOSUB 610
940 PUT SPRITE 1,(24,176),11,0
950 PUT SPRITE 2,(44,153),11,1
960 PUT SPRITE 3,(64,176),11,2
970 PUTSPRITE4,(84,153),11,3
980 PUTSPRITE5,(104,176),11,4
990 PUTSPRITE6,(124,153),11,5
1000 PUTSPRITE7,(144,176),11,6
1010 PUTSPRITE8,(164,153),11,7
1020 LOCATE 20,1,0:PRINT"┌──────┐"
1030 LOCATE 20,2,0:PRINT" PUNTOS "
1040 LOCATE 20,3,0:PRINT"└──────┘"
1050 LOCATE 20,7,0:PRINT" ______ "
1060 GOTO 1130
1070 LOCATEX,Y,0:PRINTSTRING$(2,"■"):LOCATEX,Y+1,0:PRINTSTRING$(2,"■"):RETURN
1080 FORY=0TO17STEP17:LOCATE0,Y,0:PRINTSTRING$(18,"*"):NEXTY
1090 FORY=1TO16
1100 FORX=0TO17STEP17:LOCATEX,Y,0:PRINT"*"
1110 NEXTX:NEXTY
1120 LOCATE0,20,0:RETURN
1130 '
1140 '>>SOLICITA<<
1150 '>>    Y    <<
```

```
1160 '>COMPRUEBA <
1170 '>> DATOS  <<
1180 '
1190 N=0
1200 FOR M=N TO7
1210 Z=9+M
1220 LOCATE22,Z,0:GOSUB 1830:LNA=H:PRINT LNA
1230 LOCATE24,Z,0:PRINT"-"
1240 LOCATE24,Z,0:GOSUB 1830:CLN=H:PRINT CLN
1250 X=8*(2*LNA+1):Y=8*(2*CLN-1)
1260 GOSUB 1450
1270 IFA(LNA,CLN)=9 THEN 2090
1280 FOR N=1 TO 8
1290 A(LNA,N)=9
1300 A(N,CLN)=9
1310 IFLNA-N<1 OR CLN-N<1 THEN 1330
1320 A(LNA-N,CLN-N)=9
1330 IF LNA+N>8 OR CLN+N>8 THEN 1350
1340 A(LNA+N,CLN+N)=9
1350 IF LNA+N>8 OR CLN-N<1 THEN 1370
1360 A(LNA+N,CLN-N)=9
1370 IF LNA-N<1 OR CLN+N>8 THEN 1390
1380 A(LNA-N,CLN+N)=9
1390 NEXT N
1400 GOSUB 1660
1410 NEXT M
1420 GOTO 1710
1430 END
1440 '
1450 '>> SITUA RINAS EN CASILLAS <<
1460 '
1470 ON M GOSUB 1510,1520,1530,1540,1550,1560,1570,1580
1480 GOSUB 1590
1490 RETURN
1500 END
1510 V=24:RETURN
1520 V=44:RETURN
1530 V=64:RETURN
1540 V=84:RETURN
1550 V=104:RETURN
1560 V=124:RETURN
1570 V=144:RETURN
1580 V=164:RETURN
1590 'MOVER REINAS
1600 FORW=176TOYSTEP-1:GOSUB 1640:NEXTW
1610 IFX=V THEN RETURN
1620 IFX>V THEN FORJ=VTOX:GOSUB1640: NEXT J:RETURN
1630 IFX<V THEN FORJ=VTOX STEP-1:GOSUB1640:NEXTJ:RETURN
1640 PUTSPRITE M+1,(J,W),6 ,M:RETURN
1650 '
1660 '>>>>>>> PUNTUACION <<<<<<<
1670 '
1680 P=(M*100)+P
1690 LOCATE20,6,0:PRINTUSING"######";P:RETURN
1700 '
1710 '>>> SUBRUTINA GANADOR <<<
1720 '
1730 A$="T185S2O4L64GR64GR64GR5GR64GR64GR5L4GO5R5CR5DR5ER5O4L64GR64GR64GR5GR64GR
64GR5O5L4ER5DR5O4BR5G"
1740 PLAY A$
```

```
1750 LOCATE 0,20,0:PRINT"¡ FELICIDADES! encontro >1<":PRINT
1760 LOCATE5,22,0:PRINT"solucion"
1770 FOR I=0 TO 12
1780 LOCATE14,22,0:PRINTSTRING$(12,32):FORJ=0 TO 90:NEXTJ
1790 LOCATE15,22,0:PRINT">>HAY 60 <<"
1800 FORK=0 TO 100:NEXT K:NEXTI
1810 GOTO 2270
1820 END
1830 '
1840 '>>> COMPROBACION DE ENTRADA <<<
1850 ' >>>>>> DE DATOS <<<<<<
1860 '
1870 C$=INKEY$:IFC$="" THEN 1870
1880 FOR I=1 TO 8
1890 READ B$
1900 IFB$=C$ THEN H=VAL(B$):RESTORE1950:RETURN
1910 NEXTI:RESTORE1950
1920 IFM=0 THEN N=0 ELSE N=M
1930 GOTO 1200
1940 END
1950 DATA 1,2,3,4,5,6,7,8
1960 '
1970 '>> RUTINA DE [CTRL+STOP] <<
1980 '
1990 CLS:SCREEN0:COLOR 15,4,4
2000 FOR I=0 TO15
2010 LOCATE 0,10,0:PRINTSTRING$(36,32):LOCATE0,12,0:PRINTSTRING$(36,32)
2020 FORJ=0 TO 90:NEXTJ
2030 LOCATE9,10,0:PRINT"HA ROTO EL PROGAMA"
2040 LOCATE6,12,0:PRINT"HA DE VOLVER HA COMENZAR"
2050 FORK=0 TO 150:NEXTK
2060 NEXTI
2070 GOTO 2270
2080 '
2090 '>>> SUBRUTINA DE FALLO <<<
2100 '
2110 S$="T12004AG+A1GL9FEDC+2D2"
2120 T$="T12003AG+A1GL9FEDC+2D2"
2130 PLAY S$,T$
2140 FOR I=0 TO12
2150 LOCATE 0,0,0:PRINTSTRING$(26,32)
2160 FOR J=0 TO 80:NEXTJ
2170 LOCATE 5,0,0:PRINT"INTENTELO OTRA VEZ"
2180 FOR K=0 TO 100:NEXTK
2190 NEXT I
2200 LOCATE0,0,0:PRINTSTRING$(27,32)
2210 LOCATE 5,0,0:PRINT"PRESIONE [C]"
2220 K$=INKEY$:IF K$<>"C"THEN 2220
2230 CLS:COLOR15,4,4:SCREEN0
2240 '
2250 '>> FINAL O VOLVER A EMPEZAR <<
2260 '
2270 FOR Y=20 TO 22
2280 LOCATE 0,Y,0:PRINTSTRING$(28,32)
2290 NEXT Y
2300 LOCATE 0,20,0:PRINT"PULSE [0] PARA ACABAR"
2310 LOCATE 0,22,0:PRINT"PULSE [C] PARA CONTINUAR"
2320 K$=INKEY$:IF K$="" THEN 2320
2330 IF K$="0" THEN CLS:ERASE A:P=0:END
2340 IF K$<>"C" THEN 2320
2350 ERASE A:P=0:CLS:GOTO770
```

Calendario

Alguna vez habrás sentido curiosidad por saber en qué fecha caía un cumpleaños, o alguna fiesta importante, que nos permita tomarnos un pequeño puente. Sí no teneís un calendario a mano, conectar el ordenador e introducir este programa con él podreís obtener el calendario de cualquier mes a partir del año 1752, en adelante. Una vez introducido y ejecutado el programa, aparecerá en la pantalla el calendario del mes en cuestión y tres opciones que te permitirán ojear el calendario del mes anterior, el mes siguiente y abandonar el programa. Con ellas podremos ver las fechas pasadas y las venideras sin tener que ejecutar nuevamente el programa. Para aquellos que posean impresora. les vendría bien adaptar el programa de forma que se pidieran listar los meses escogidos y así tenerlos a mano cuando lo deseemos.




```
10 WIDTH 29: COLOR 15,5,5:CLS
20 M$="EneFebMarAbrMayJunJulAgoSepOctNovDic"
30  N$="312831303130313130313031"
40 INPUT "Año (>1752)";Y: IF Y<1753 THEN PRINT "De eso hace mucho tiempo, se más
 moderno": GOTO 40
50 INPUT "Mes (1-12)";M: IF M<1 OR M>12 THEN PRINT "Conoces tu ese mes, listo ?"
:GOTO 50
60 CLS : LOCATE 8,4:PRINT MID$(M$,3*M-2,3); TAB(18);Y
70 LOCATE  2,7:PRINT "Dom Lun Mar Mie Jue Vie Sab"
80 MID$(N$,3,4)="28"
90 IF Y=4*INT (Y/4) AND Y<>100*INT (Y/100) THEN MID$(N$,3,4)="29"
100 D=3+Y+INT ((Y-1)/4)-INT ((Y-1)/100)
110 FOR A=1 TO M-1: LET D=D+VAL(MID$(N$,2*A-1,2)): NEXT A
120 D=D-7*INT (D/7)
130 LOCATE 0,9
140  FOR A=1 TO VAL(MID$(N$,2*M-1,2))
150  P=2+4*(A+D-7*INT ((A+D)/7))
160 PRINT TAB(P);A;
170 NEXT A
180 LOCATE 0,20:PRINT"'n'-mes siguiente":PRINT"'p'-mes anterior ":PRINT"'q'-sali
r del programa"
190 I$=INKEY$
200 IF I$<>"n" AND I$<>"q" AND I$<>"p" THEN GOTO 190
210 IF I$="n" THEN  M=M+1
220 IF M>12 THEN  M=1:  Y=Y+1
230 IF I$="p" THEN M=M-1
240 IF M<1 THEN M=12: LET Y=Y-1
250 IF I$<>"q" THEN GOTO 60
```

Ecuaciones de primer y segundo grado

Esperemos que a lo largo de todo el año hayaís sido muy estudiosos, puesto que en verano, no es cuestión de encerrarse y estudiar. Ahora bien, si habéis tenido mala suerte o no pudísteis hacer esa labor como corresponde, esta es la oportunidad que necesitábais para repasar la materia.

Por el contrario, si habéis salido bien parados de las notas de fin de curso, probablemente utilicéis el ordenador para jugar y pasar el rato.

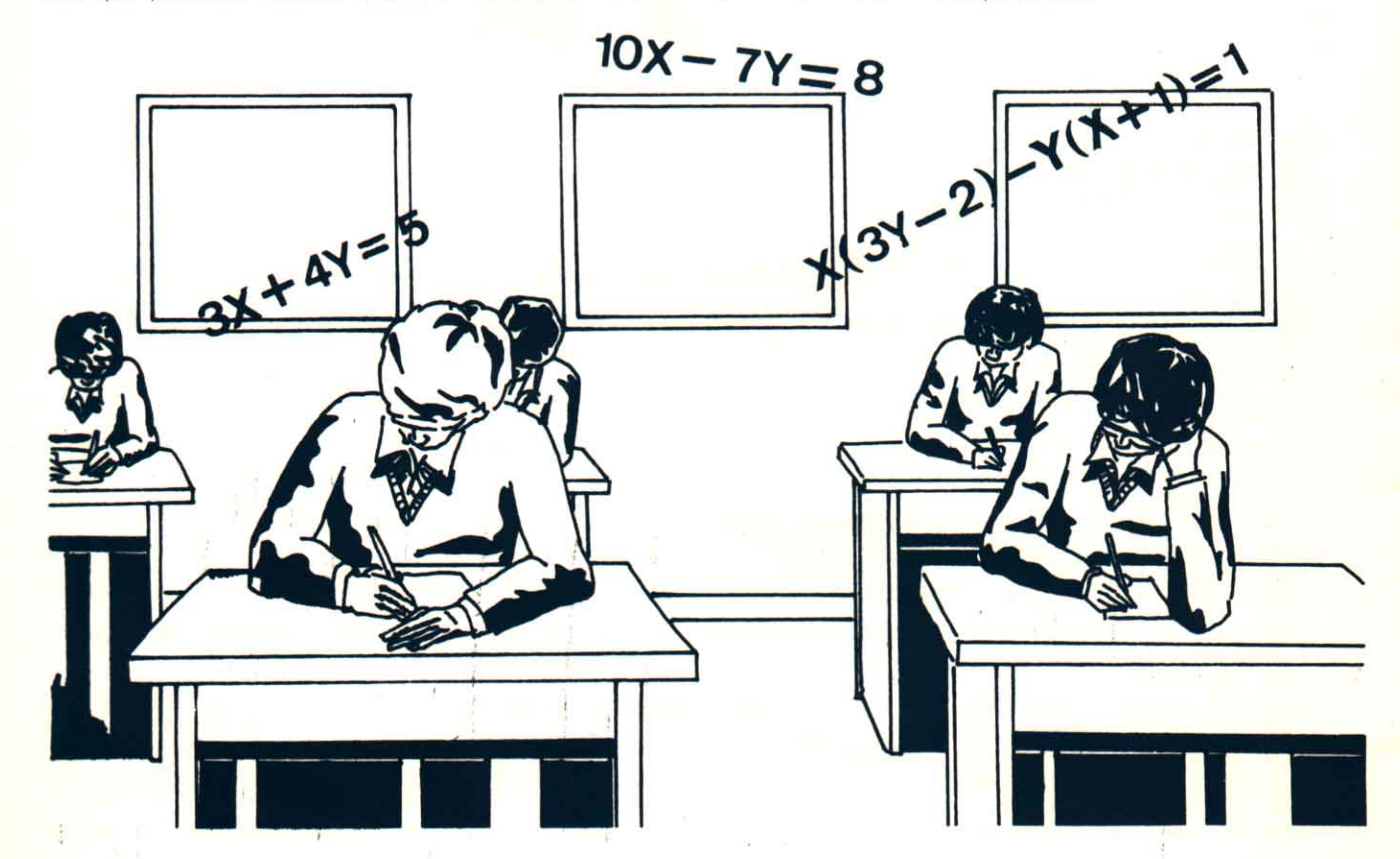
Sin embargo, hayáis o no aprobado, este programa os hará repasar y comprobar vuestro conocimientos.

El programa presenta unas opciones que permiten resolver ecuaciones de primer y segundo grado, con la particularidad de poder interrogar al ordenador de la misma forma que éste te preguntará a ti. De esa manera, sabréis hasta qué punto llegan vuestros conocimientos.

Al ejecutarlo, aparecerán las dos opciones antes especificadas, si escogemos la primera, el ordenador nos preguntará por el tipo de ecuación a resolver, que pueden ser de una o dos incógnitas. Inmediatamente después se seleccionan los coeficientes de la ecuación de forma totalmente aleatorio. En la mayoría de los casos los números son reales, es decir, con infinitos números decimales. Esto no deberá preocuparos, ya que el programa está preparado para obtener la solución con 4 ó 5 decimales, algo bastante corriente.

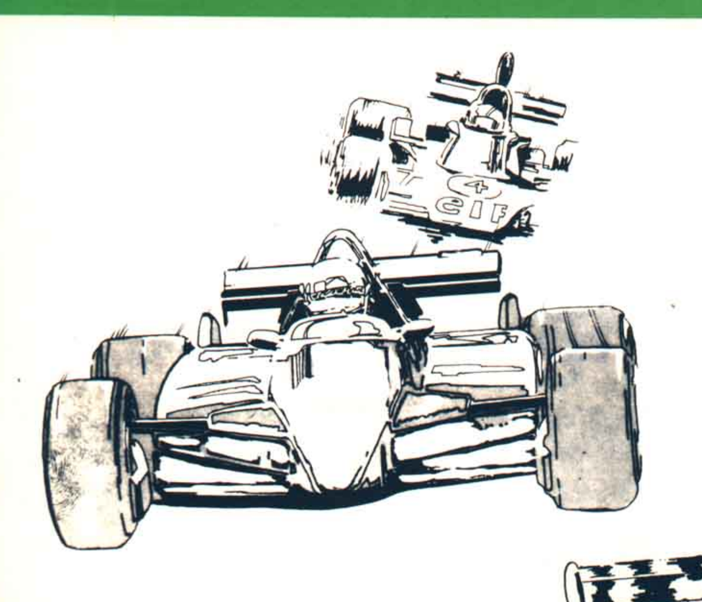
No obstante, si no queréis que el ordenador pueda poneros en evidencia (no se porta mal con los que fallan), podéis elegir la opción 2, en la que vosotros indicáis los coeficientes de la ecuación y el ordenador os muestra la solución, con la particularidad de que en ecuaciones de segundo grado también da soluciones complejas.

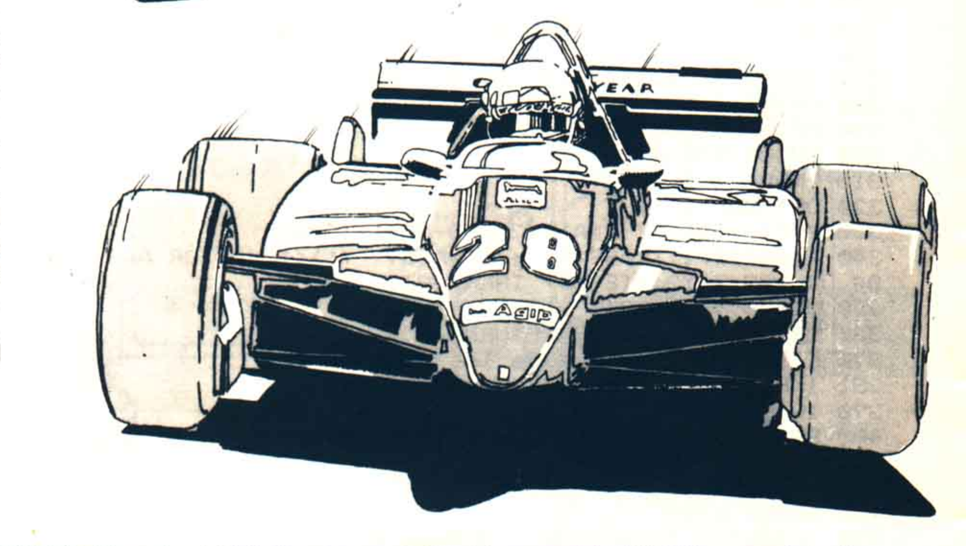
Esperamos que demostréis a vuestra máquina lo listos que sois y seais vosotros los que la pongais en compromisos.



```
10 KEY OFF: CLS
20 LOCATE 10,10: PRINT "ECUACIONES"
30 LOCATE 0,21:PRINT "Pulsa una tecla"
40 IF INKEY$="" THEN GOTO 40
50 CLS:PRINT  TAB(5);"MENU": LOCATE 0,2:PRINT "1- el ordenador pregunta":PRINT "
2- tu preguntas"
60 LOCATE 0,21: INPUT "Elige"; OP: IF OP<1 OR OP>2 THEN GOTO 60
70 LOCATE 0,5:PRINT :PRINT "Ecuacion de primer grado":PRINT :PRINT "Ecuacion de
segundo grado"
80 LOCATE 0,21:INPUT "elige";EC:CLS
90 IF OP=2 THEN GOTO 370
100 FOR N=1 TO 3
110 A(N)=INT (RND(-TIME)*50)-25:NEXT N:  GOSUB 360
120 A=A(1):B=A(2):C=A(3): IF A=0 OR DET<0  THEN GOTO 100
130 IF EC=2 THEN GOTO 240
140 GOSUB 540
150 PRINT :PRINT A;"x";:IF B>=0 THEN PRINT "+";
160 PRINT B;"=";C
170 GOSUB 550: INPUT X: IF ABS( X-((C-B)/A))< 1E-04THEN GOTO 190
180 GOSUB 530: GOTO 140
190 LOCATE 0,19: PRINT "Muy bién, lo has sabido contestar,   si quieres volver a
 hacer otra pul-  sa 'o', si no, pulsa 's' "
200 I$=INKEY$
210 IF I$="s" OR I$="S"  THEN END
220 IF I$="o" OR I$="O"  THEN RUN
230 GOTO 200
240 GOSUB 540
250 GOSUB 360  : IF DET<0 THEN GOTO 100
260 LOCATE 1,5:PRINT "Cual es la solucion de la ecuacion:":PRINT A;"x";
270 PRINT CHR$(253);:IF B>=0 THEN PRINT "+";
280 PRINT B;"X"; : IF C>=0 THEN PRINT "+";
290 PRINT C;"=0"
300 GOSUB 550
310 INPUT X,Z
320 DET=SQR(DET)
330 Y1=(-B+DET)/(2*A):Y2=(-B-DET)/(2*A)
340 X=ABS(X):Y1=ABS(Y1): IF(ABS(X-Y1)<1E-04 OR ABS(X-Y2)<1E-04)AND(ABS(Z-Y1)<1E-
04 OR ABS(Z-Y2)<1E-04) THEN GOTO 190
350 GOSUB 530: GOTO 310
360 DET=B^2-(4*A*C): RETURN
370 CLS:PRINT "introduce los datos"
380 INPUT"a";A:INPUT"b";B:INPUT"c";C
390 IF EC=2 THEN GOTO 420
400 PRINT "La solucion para la ecuacion ":PRINT A;"x+";B;"=";C: PRINT "es :":PRI
NT "x=";(C-B)/A
410 GOTO 500
420 GOSUB 360
430 IF DET<0 THEN COM=1: DET=DET*(-1)
440 DET=SQR(DET)
450 PRINT "La solucion es":IF COM=1 THEN GOTO 480
460 PRINT "x=";(-B-DET)/2*A:PRINT "x=";(DET-B)/2*A
470 GOTO 500
480 PRINT "x=";-B/(2*A);"+";DET/(2*A);"i": PRINT "x=";-B/(A*2);"-";DET/(2*A);"i"
490 GOTO 500
500 PRINT " si quieres volver al menu, pulsa 'm'si no pulsa 's':
510 INPUT S$: IF S$="m" THEN RUN
520 IF S$="s" THEN END ELSE GOTO 510
530 PRINT  "Lo siento pero no es cierto. Vuelve aintentarlo": RETURN
540 LOCATE 1,5: PRINT "Cual es la solución de la ecuacion:": RETURN
550 LOCATE 0,16: PRINT " con 4 ó 5 decimales vale ": RETURN
```

Jarama



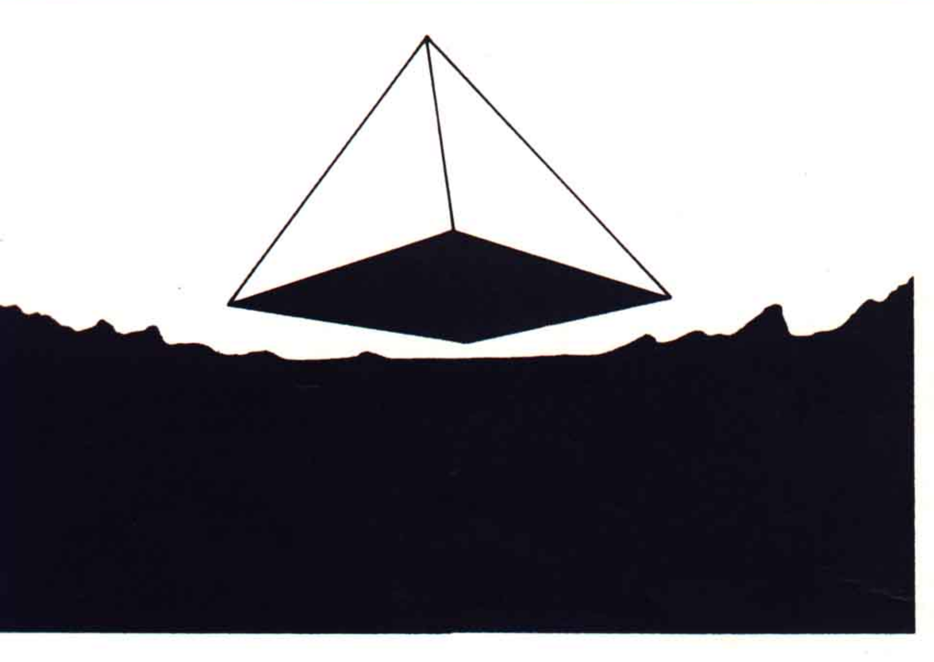


Si algún día sientes la necesidad de hacer el loco en coche sin salir de casa, "enchúfate" al MSX y siente el gusanillo de la velocidad. El juego consiste en un coche que ha de recorrer el circuito poniendo especial atención en las curvas que irán apareciendo sin cesar. Estas, se hallan protegidas por neumáticos de manera que si chocas contra uno de ellos te estrellarás, finalizando así el intento de lograr el record. Además a consecuencia del coche aparecerán los coches escoba, que recogerán tus restos. Cada 200 kilómetros que superes, aparecerá una meta. ¿Hasta cúal de ellas crees que llegarás?...

José María Cavanillas (Madrid)

```
10 REM ****************
20 REM * JARAMA, DE    *
30 REM * J.M.C. PARA   *
40 REM * MSX MAGAZINE *
50 REM ****************
60 PLAY"T120V8M255S13R1R2O4L16E4FGO5C2O4D4EF2R4F4GAO5D2O4D4EF4G4A4R8E4FGO5C2D4EF
2"
70 TIME=0:FORT=1TO32:L=INT(T/8):READQ:A$(L)=A$(L)+CHR$(Q):NEXTT
80 FORT=1TO32:L=INT(T/8):READQ:B$(L)=B$(L)+CHR$(Q):NEXTT
90 FORT=1TO32:L=INT(T/8):READQ:C$(L)=C$(L)+CHR$(Q):NEXTT
100 COLOR 6,15,2:SCREEN3,2:LINE(0,0)-(20,191),2,BF:LINE(240,0)-(255,191),2,BF:
110 FORT=32TO224STEP8:VPOKET,31:VPOKET+1,241:NEXT:FORT=24TO1304STEP256:FORR=TTOT
```

```
+6STEP2:VPOKER,31:VPOKER+1,241:NEXT:NEXT
120 FORT=232TO1512STEP256:FORR=TTOT+6STEP2:VPOKER,31:VPOKER+1,241:NEXT:NEXT:FORT
=1318TO1510STEP8:VPOKET,31:VPOKET+1,241:NEXT
130 OPEN"GRP:"AS#1:FORT=150TO20STEP-8:LINE(40,T+1)-(230,T+32),15,BF:PRESET(40,T)
:PRINT#1,"JARAMA":NEXT:COLOR 9
140 SPRITE$(0)=A$(0)+A$(1)+A$(2)+A$(3):SPRITE$(1)=B$(0)+B$(1)+B$(2)+B$(3):SPRITE
$(2)=C$(0)+C$(1)+C$(2)+C$(3):FORT=39TO165:PUTSPRITE0,(77,T),6:PUTSPRITE1,(141,T)
,2,0:PUTSPRITE2,(205,T),4,0
150 IFT=130THENPRESET(90,105):PRINT#1,"POR"
160 NEXT:FORT=0TO2:PUTSPRITET,(0,0),0,4:NEXT:PRESET(40,150):PRINT#1,"J.M.C."
170 IFTIME<700GOTO 170
180 COLOR1,15,2:SCREEN2:LINE(0,0)-(15,191),2,BF:LINE(248,0)-(255,191),2,BF:FORT=
6146TO6174:VPOKET,2:NEXT:FORT=6178TO6850STEP32:VPOKET,2:VPOKET+28,2:NEXT:FORT=68
82TO6910:VPOKET,2:NEXT
190 LINE(16,0)-(19,3),1,BF:LINE(20,4)-(23,7),1,BF:LINE(16,64)-(19,67),1,BF:LINE(
20,68)-(23,71),1,BF:LINE(16,128)-(19,131),1,BF:LINE(20,132)-(23,135),1,BF
200 COLOR 8:PRESET(100,16):PRINT#1,"JARAMA":PRESET(50,50):PRINT#1,"1.- TECLAS DE
 CURSOR.":PRESET(50,70):PRINT#1,"2.- PALANCA DE MANDOS."
210 A$=INKEY$:R=RND(1):IFA$=""GOTO 210
220 IFA$<>"1"ANDA$<>"2"GOTO 210ELSEQ=VAL(A$)-1
230 DIMA(4):LINE(50,16)-(220,78),15,BF:COLOR 1
240 FORT=0TO4:A(T)=10:NEXTT:X=120:TIME=0
250 FORT=0TO4:LINE(INT(A(T))*8,T*8+8)-(INT(A(T))*8+87,T*8+15),15,BF:CIRCLE(INT(A
(T))*8+20,T*8+12),2:LINE(INT(A(T))*8+24,T*8+8)-(INT(A(T))*8+63,T*8+15),14,BF:CIR
CLE(INT(A(T))*8+67,T*8+12),2:NEXT
260 IFTIME>11200THENCK=CK+1:GOTO 380
270 FORT=0TO3:A(T)=A(T+1):NEXT:A(4)=A(4)+RND(1)*3-1.5:IFA(4)<2THENA(4)=2
280 IFA(4)>20THENA(4)=20
290 IFSTICK(Q)=3THENX=X+8
300 IFSTICK(Q)=7THENX=X-8
310 IFPOINT(X,24)<>14ORPOINT(X+8,24)<>14ORPOINT(X,39)<>14THEN330
320 PUTSPRITE0,(X,25),4,0:GOTO 250
330 SOUND7,247:SOUND8,15:FORN=0TO30:SOUND6,N:PUTSPRITE0,(X,25),9,1:FORJ=1TO20:NE
XT:PUTSPRITE0,(X,25),8,2:FORJ=1TO10:NEXT:NEXT:PUTSPRITE0,(X,25),4,2:SOUND8,0:SOU
ND7,254
340 COLOR 6:PRESET(32,70):PRINT#1,"TE HAS SALIDO DE LA PISTA":PRESET(32,90):PRIN
T#1,"DESPUES DE HABER REBASADO":PRESET(32,110):PRINT#1,"LOS";CK*200+INT(TIME/56)
;"KILOMETROS."
350 PRESET(32,130):PRINT#1,"PARA JUGAR DE NUEVO,PULSA":PRESET(32,150):PRINT#1,"L
A BARRA ESPACIADORA."
360 IFSTRIG(0)=0THENR=RND(1):GOTO 360
370 COLOR 1:CK=0:GOTO430
380 LINE(INT(A(4))*8+16,48)-(INT(A(4))*8+23,63),9,BF:LINE(INT(A(4))*8+24,48)-(IN
T(A(4))*8+63,63),10,BF:LINE(INT(A(4))*8+64,48)-(INT(A(4))*8+71,63),9,BF
390 PRESET(INT(A(4))*8+30,52):PRINT#1,"META":PLAY"T120V8M255S13L16O4GR64GR64GO5E
4D8O4GO5E4D8O4GO5E4D4C2":FORT=32TO160:PUTSPRITE0,(INT(A(4))*8+40,T),4,0:NEXT:COL
OR 6
400 PRESET(34,70):PRINT#1,"HAS REBASADO LOS PRIMEROS":PRESET(26,90):PRINT#1,CK*2
00;"KILOMETROS.":PRESET(34,110):PRINT#1,"PARA SEGUIR EN LA CARRERA":PRESET(34,13
0):PRINT#1,"PUEDES PRESIONAR LA BARRA":PRESET(34,150):PRINT#1,"DE ESPACIO."
410 COLOR 1:PUTSPRITE0,(0,0),0,0
420 IFSTRIG(0)=0GOTO420
430 FORT=8TO160STEP2:PUTSPRITE0,(77,T),6,0:PUTSPRITE1,(141,T),2,0:PUTSPRITE2,(20
5,T),4,0:LINE(24,T-1)-(239,T),15,BF:NEXT:PUTSPRITE0,(0,0),0:PUTSPRITE1,(0,0),0:P
UTSPRITE2,(0,0),0:GOTO240
440 DATA255,255,17,224,224,224,32,42,36,17,10,59,59,10,4,63,224,224,0,224,224,22
4,128,128,128,0,0,128,128,0,0,128
450 DATA0,0,0,0,6,9,0,32,0,16,0,9,6,0,0,0,0,0,0,0,0,0,128,0,128,0,0,0,0,0,0
460 DATA128,8,64,2,0,32,0,0,2,0,32,0,2,64,8,128,64,0,0,0,0,32,0,0,0,0,0,0,0,0,
0
```



Triángulo mágico

No todos los programas van a ser de invasiones extraterrestres, también hay programas que de por sí. nos complican la vida.

Más de un lector recordará la época del cubo de Rubik, pues bien, este programa no pretende sustituir el cubo maágico, más bien es como una herramienta que nos permitirá desarrollar nuestra capacidad mental, para frontar el reto de resolver el problema del cubo.

En él, nos aparecerá un triángulo formado por grupos de letras desordenadas al azar y números, gracias a los cuales podemos rotar los grupos de letras, como explicaremos a continuación.

El movimiento es cíclico y sólo afecta a las tres letras que rodean al número que escogamos. Este movmiento es a derechas, pudiéndo realizarlo tantas veces como nosotros queramos, teniéndo en cuenta, que girar más veces de las debidas, es hacerlo sobre sí mismo. Como es obvio, nuestro objetivo es intentar ordenar alfabéticamente las letras de arriba a abajo.

Una vez que tengamos la experiencia necesaria, podremos intentarlo resolver, colocando las letras en orden alfabético en el orden inverso. Este caso, sólo es recomendado para lo más expertos.

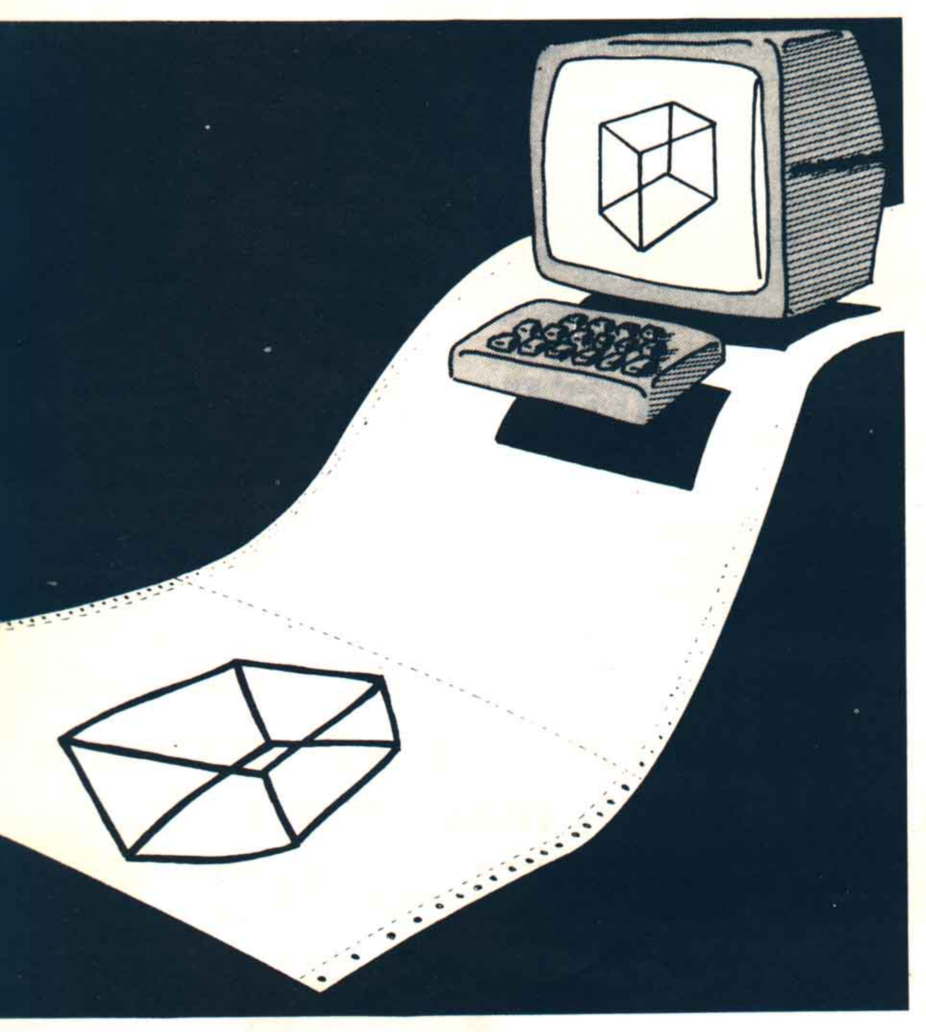
De cualquier manera y aunque parezca difícil, no lo es tanto, que nadie piense que es coser y cantar, ya que ordenar las últimas letras, con las que a veces no se cuenta, es de por sí un gran problema y nos causarán más de un dolor de cabeza.

Una vez completado el triángulo, el ordenador nos felicitará no muy efusivamente, y nos retará intentarlo otra vez, con letras desordenadas.

```
5 RDRST=RND(-TIME)
10 REM TRIANGULO
20 A$="abcdefghij"
30 CLS:KEY OFF:WIDTH 35
40 LOCATE 0,2: PRINT;"Dentro de un momento va a aparecer un triangulo con letras
 desordena- das al azar. Hay que intentar orde narlas moviendo los bloques de le
- tras que rodean a un numero. El mo vimiento es a derechas.          "
50 PRINT,,,,"Cuando estes listo pulsa una tecla"
60 IF INKEY$="" THEN GOTO 60 ELSE CLS
100 REM GENERACION DE CARACTERES
110 DIM B$(10):C=0
120 FOR N=1 TO 10
130 C=1+INT (RND(1)*10)
140 IF N=1 THEN GOTO 170
150 FOR L=N-1 TO 1 STEP -1: IF B$(L)=CHR$(64+C) THEN GOTO 130 ELSE NEXT L
170 LET B$(N)=CHR$ (C+64)
180 NEXT N
190 GOSUB 200:GOTO 1000
200 REM PRESENTACION DE PANTALLA
205 RESTORE 900
210 FOR N=1 TO 10
220 READ X,Y:IF N>6 THEN GOTO 250
230 LOCATE  Y,(X+1)  : PRINT CHR$ (48+N)
250 LOCATE Y,X:PRINT B$(N)
260 NEXT N
```

```
270 RETURN
900 DATA 4,4,6,3,6,5,8,2,8,4,8,6,10,1,10,3,10,5,10,7
1000 REM INTRODUCIR MOVIMIENTO1 nu-1010 LOCATE 0,15: INPUT"introduce el grupo a
girar y el nu-mero de veces que tiene que girar ";S$ :if len(s$)=1 then goto 101
0
1010 LOCATE 0,15: INPUT"introduce el grupo a girar y el nu-mero de veces que tie
ne que girar           ";S$
1020 G1=VAL(LEFT$(S$,1)):V=VAL(MID$(S$,2,3)):G=G1
1030 IF G=1 THEN LET G2=2: GOTO 1100
1040 IF G<4 THEN LET G2=G+2:GOTO 1100
1050 G2=G+3
1100 G3=G2+1
1110 FOR N=1 TO V
1120 X$=B$(G):B$(G)=B$(G2):B$(G2)=B$(G3):B$(G3)=X$
1130 NEXT N
1140 GOSUB1500
1150 IF LEN(S$)>3 THEN S$=MID$(S$,3):GOTO 1020
1160 GOTO 1010
1500 IF B$=A$ THEN GOTO 2000
1510 GOSUB 200:RETURN
2000 LOCATE0,20:PRINT "Muy bien, lo has conseguido. Si te atreves a intentarlo o
tra vez pulsa'p', si no 's'"
2010 A$=INKEY$ :IF A$="" THEN GOTO 2010
2020 IF A$="s" THEN STOP ELSE RUN
```

Plotter MSX



Un programa que no podía faltar en un ordenador es un plotter. Este es auto explicativo, ya que viene con instrucciones completas, que aconsejamos se estudien bien, para sacar el máximo provecho de él. A ejecutarlo aparecerán las instruciones ya mencionadas y nos preguntará por el modo de pantalla y si utilizamos *josticks* o las teclas del cursor.

Una vez contestadas estas dos preguntas, se nos presentarán cinco opciones que son las de dibujar con cursores, dibujar círculos, borrar la pantalla, cambiar los colores y modo de textos.

En la opción 1, dibujaremos con un cursor que aparecerá y nos permitirá movernos en ocho direcciones con las teclas del cursor.

Para dibujar tendremos que bajar la pluma con la barra espaciadora o el botón de disparo. Una vez hecho esto, podremos dibujar.

Resulta interesante estudiar un poco el programa, ya que utiliza las teclas de función para realizar una serie de operaciones muy interesantes. Así. tenemos que la tecla F5/F10, nos permitirá borrar (operación que se realiza de la misma forma en que dibujamos). Las teclas F2/F7 nos permitirán unir dos puntos cualesquiera. Para ello, debemos desplazar la pluma hasta el primer punto y fijarlo con la tecla F2/F7, a continuación desplazaremos la pluma hasta el punto 2 y pulsaremos F3/F8. Por último, y dentro de esta opción, podremos rellenar un área con sólo posicionar la pluma dentro de la zona deseada y pulsando F4/F9.

En la opción 2, podremos dibujar círculos, arcos, elipses y parábolas.

En esta opción debemos especificar los siguientes parámetros, el ángulo inicial y final (en radianes) yel radio (en *píxels*). Para ello trazar círculos completos especificar 0 y 6.28 para el ángulo inicial y final, respectivamente.

A continuación tenemos la opción 3, que, como su propio nombre indica, su único propósito es el de borrar la pantalla.

En la siguiente opción podremos cambiar los colores de la pantalla tal y como nosotros queramos.

Por último, la opción 5, que es el modo de textos, permite incluir textos y caracteres gráficos en la pantala. Estos se escribirán a partir de la última coordenada dibujada.

Para volver al menú desde cualquier punto del programa, pulsaremos F1.

Para concluir, en la esquina inferior derecha aparecerán unas letras que nos indican el estado de la pluma en cada momento, estos son (MB), cuando la pluma se encuentre en modo borrado. MD cuando ésta se encuentre en modo de dibujo y MP cuando la pluma se encuentra levantada (no dibuja).

Ahora sólo queda practicar y realizar todos aquellos dibujos que tanto deseabas tener.

```
10 REM PLOTTER
11 REM
20 REM O.L.V. SOFT 1985
21 REM
30 REM MSX MAGAZINE
40  GOSUB 590
50 SCREEN 1,2:SPRITE$(0)=CHR$(128)
60 RESTORE 700:FOR A=1 TO 3:FOR B=1 TO 32:READX:B$=B$+CHR$(X):NEXTB:SPRITE$(A)=B$:B$="":NEXT A
70 RESTORE
80 CD=15:PE=0:P=4:B=7:X=128:Y=91:PB=4:CLS
90 LOCATE 2,10:PRINT"TECLADO O JOYSTICK (T/J)"
100 C$=INKEY$:IF C$="J"THEN JY=1ELSE IFC$="T"THEN JY=0:ELSE 90
110 KEYOFF
120 FOR A=38900!TO 38925!:READ C:POKE A,C:NEXT:DEFUSR=38900!:DEFUSR1=38913!:GOSUB 580
130 ON KEY GOSUB 520,410,420,470,680:KEY(1)ON:KEY(2)ON:KEY(3)ON:KEY(4)ON:KEY(5)ON
140 OPEN "GRP:"AS#1
150 GOSUB 460
160 INPUT "MODO DE PANTALLA (2 O 3)";SC:IF SC<>2AND SC<>3 THEN 160
170 GOSUB 490
180 ON STRIG GOSUB 540,540,540,470:STRIG(0)ON
190 STRIG(1)ON:STRIG(3)ON
200 CLS:COLOR 15,4,4
210 RESTORE 570 :COLOR 15,4,7:FOR A=2 TO 14STEP 3:READ A$:LOCATE 2,A:PRINTA$:NEXT
220 LOCATE 5,20:LINE INPUT"INTRODUZCA OPCION";OP$:OP=VAL(OP$):IF OP>5 OR OP<1 THEN 220
230 ON OP GOTO 240,430,480,500,510
240 SCREEN SC:COLOR CD,P,B:IF C>=1THENA=USR1(A)
250 D=STICK (JY)
260 IF D=1 THENY=Y-1
270 IF D=3 THENX=X+1
280 IF D=5 THENY=Y+1
290 IF D=7 THENX=X-1
300 IF D=2 THENX=X+1:Y=Y-1
310 IF D=4 THENX=X+1:Y=Y+1
320 IF D=6 THENX=X-1:Y=Y+1
330 IF D=8 THENX=X-1:Y=Y-1
340 PUT SPRITE 0,(X,Y),15,0
350 IF PE=1 THEN PUT SPRITE 2,(240,175),CD,1
360 IF PE=0 THEN PUT SPRITE 2,(240,175),CD,2
370 IF CB=1 THEN PUT SPRITE 3,(240,0),CD,3ELSE PUT SPRITE 3,(0,192)
380 IF CB=1THEN PUT SPRITE 2,(240,175),15,2:PSET(X,Y),PB:GOTO 250:'BORRADO
390 IF PE=1 THEN PSET (X,Y),CD
400 GOTO 250
410 XX=X:YY=Y:PSET(X,Y),CD:RETURN
420 X1=X:Y1=Y:LINE (XX,YY)-(X,Y1),CD:RETURN
430 SCREEN 1:INPUT "RADIO";R:INPUT"ANGULO INICIAL";AI:INPUT "ANGULO FINAL";AF:INPUT "FACTOR DE CORRECCION";FC:INPUT "COLOR";CC
440 SCREEN SC:A=USR1(A)
450 CIRCLE (X,Y),R,CC,AI,AF,FC:GOTO 250
460 SCREEN SC:CLS:A=USR(A):RETURN
470 PAINT(X,Y),CD:RETURN
480 SCREEN SC:COLOR CD,P,B:PB=P:CLS:A=USR(A):GOTO 240
490 SCREEN SC:A=USR(A):SCREEN 1:RETURN
500 SCREEN 1:INPUT"COLOR DEL FONDO";P:INPUT"COLOR DEL BORDE";B:INPUT "COLOR DE LA PLUMA";CD:SCREEN SC:A=USR1(A):COLOR CD,P,B:GOTO 250
510 CLS:LOCATE 2,6:LINE INPUT "TEXTO ";A$:SCREEN SC:A=USR1(A):PSET(X,Y),CD:COLOR CD:PRINT#1,A$:PE=0:GOTO 250
520 PUT SPRITE 2,(0,192):PUT SPRITE3,(0,192): A=USR(A):PE=0:C=1
```

```
530 SCREEN 1:RETURN 210
540 IF PE=1 THEN PE=0:ELSE PE=1
550 RETURN
560 DATA 33,0,0,17,64,156,1,0,64,205,&H59,0,201,33,64,156,17,0,0,1,0,64,205,&H5C
,0,201
570 DATA (1) DIBUJO CON CURSORES,(2) CIRCULOS,(3) BORRAR PANTALLA,(4) CAMBIAR CO
LORES,(5) TEXTO
580 SCREEN SC:A=USR(A):RETURN
590 SCREEN0:WIDTH39:LOCATE 13,1:PRINT"PLOTTER MSX"
600 LOCATE 6,3:PRINT"O.L.V. SOFT 1985"
610 LOCATE 0,6:PRINT"DIBUJO CON CURSORES":PRINT:PRINT"LA PLUMA SE DESPLAZA CON L
AS TECLAS ":PRINT:PRINT"CURSOR O CON UN JOY STICK EN EL PORT 1"
620 PRINT:PRINT"LA BARRA ESPACIADORA LEVANTA Y BAJA LA ":PRINT"PLUMA.":PRINT:PRI
NT" (F5) ACTIVA  EL BORRADO"
630 PRINT:PRINT" (F4) PAINT":PRINT:PRINT" (F1) MENU"
640 LOCATE 9,22:PRINT"PULSA UNA TECLA "
650 IF INKEY$=""THEN 650
660 RETURN
670 STOP
680 IF CB=1THEN CB=0ELSECB=1
690 RETURN
700 DATA 0,0,65,99,85,73,65,65,65,65,65,65,65,0,0,0,0,0,96,80.72,72,72,72,72,72,
72,80,96,0,0,0
710 DATA 0,0,65,99,85,73,65,65,65,65,65,65,65,65,0,0,0,0,112.72,68,68,72,112,64.
64,64,64,64,64,0,0
720 DATA 0,0,65,99,85,73,65,65,65,65,65,65,65,65,0,0,0,0,96,80,72,72,80,96,96,80
,72,72,80,96,0,0
```

LOS JUEGOS ELECTRONICOS

Programas Sony para ordenadores MSX

A la orden.


Monkey Academy


Países del Mundo-1


Países del Mundo-2


Computador Adivino


Computer Billiards


The Snowman


Cubit


Character Collection


Stop the express (Para el Tren)


Hustler (Billar Americano)


Data cartridge


Quinielas y Reducciones


Home Writer


Sparkie


Aprendiendo Inglés-1


Binary Land


Creative Greetings


Aprendiendo Inglés-2


Antartic Adventure


Mastermind


Contabilidad Personal


Athletic Land


E.I.


Ficheros


El Ahorcado


Dorodon


La Pulga


Cosmos


Control de Stocks


Battle Cross


Mouser


Crazy Train


Ali baba


Juno First


Car Jamboree


Tutor


Track and Field-1 (olimpiadas)


Blackjack


Track and Field-2 (olimpiadas)


Driller Tanks (Tanque Destructor)


Sonygraph


Ninja (El Samurai)


Les Flics

Y muchos más títulos

Ordenador Doméstico

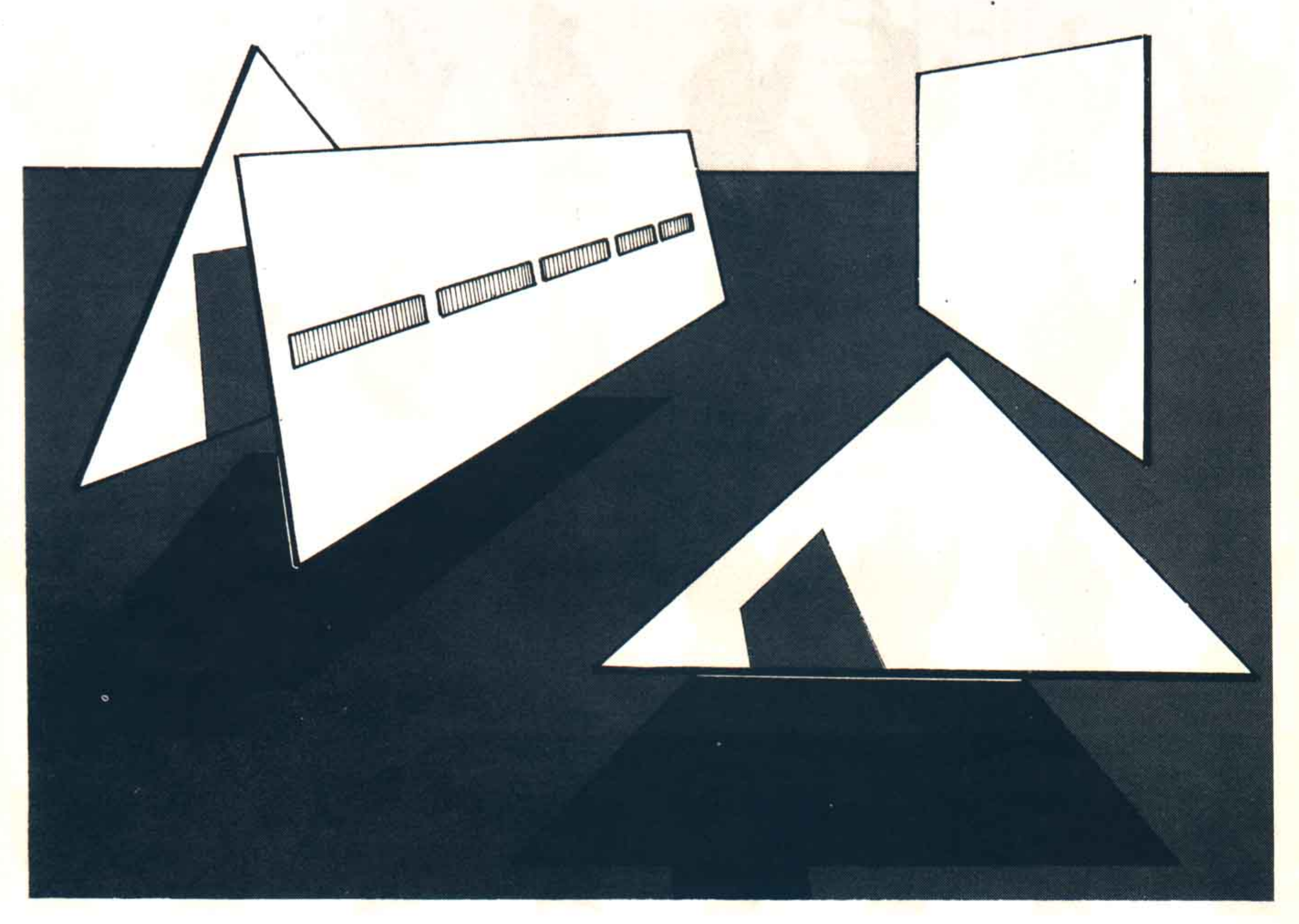




Para lo que guste ordenar.



Superficies



¿Sabría decirme algún lector ¿cuántos metros cuadrados tiene un campo de fútbol? o ¿cuántos metros cuadrados tiene una cancha de baloncesto? No es una cuestión de vida o muerte, pero alguna vez le habrán planteado problemas de este tipo. Resolverlos se logra con la aplicación de unas simples fórmulas.

De cualquier manera, el que se inicie en esta rama de las matemáticas, gustará de contar con este programa.

En el menú de opciones aparecen cuatro apartados. Podremos calcular la superficie generada por un triángulo, cuadrado, rectángulo, círculo y un arco.

Una vez elegida la opción, inmediatamente se dibujará la figura que se ha escogido. Introduciremos los datos, uno a uno y al momento obtendremos los resultados.

Es importante destacar, que el programa es simple y fácil de entender e introducir. Muy útil para todos aquellos principiantes en la materia y cuyo futuro esté en la arquitectura, ya que por algo hay que empezar.

```
10 SCREEN 0: COLOR 15,4,4:CLS:WIDTH 35: KEY OFF
20 LOCATE 12,0: PRINT "Menu":PRINT : PRINT "1- triangulo":PRINT"2- cuadrado"
30 PRINT "3- rectangulo":PRINT"4- circulo":PRINT "5- arco"
40 PRINT :INPUT "introduce el numero de opcion deseada";OP
50 SCREEN2:OPEN "grp:"AS#1
60 ON OP GOSUB 130,220,290,380,480
70 PRESET (10,150):PRINT#1,"si quieres calcular mas":PRESET (10,165):PRINT#1,"su
perficies pulsa 'p',":PRESET (10,180):PRINT #1,"si no, 's'"
80 CLOSE#1
90 I$=INKEY$
100 IF I$="s" THEN STOP
110 IF I$="p" THEN RUN
120 GOTO 90
130 REM Triangulo
140 INPUT "base ";B::PRINT B:INPUT "altura";A:PRINT A
150 SCREEN 2
160 LINE (100,100)-(200,100),5:LINE -(150,50),5:LINE -(100,100),5
170 PAINT (125,99),5
180 PRESET(10,20): PRINT #1,"base ";B;"altura ";A
190 S=(B*A)/2
200 GOSUB 590
210 RETURN
220 REM Cuadrado
230 INPUT "lado ";L:PRINT L
240 SCREEN 2
250 LINE (100,100)-(167,10),5,BF
260 PRESET(10,20):PRINT#1,"lado ";L
270 S=L^2
280 GOSUB 590:RETURN
290 REM Rectangulo
300 INPUT "lado mayor ";L: PRINT L
310 INPUT "lado menor ";LM: PRINT LM
320 SCREEN 2
330 LINE (100,100)-(200,50),5,BF
340 PRESET (10,20):PRINT #1,"lado mayor ";L
350 PRESET (10,30):PRINT #1,"lado menor ";LM
360 S=L*LM
370 GOSUB 590:RETURN
380 REM Circulo
390 PI=4* ATN(1)
400 INPUT "radio ";R:PRINT R
410 SCREEN 2
420 CIRCLE (150,60),50,5,,,1.4
430 PAINT (151,81),5
440 PRESET (10,20): PRINT #1,"radio ";R
450 S=PI*R^2
460 GOSUB 590
470 RETURN
480 REM Arco
490 INPUT "radio ";R:PRINT R
500 INPUT "angulo en radianes ";A
510 SCREEN 2
520 LINE (200,60)-(150,60),5
530 CIRCLE (150,60),70,5,0,.5236,1.4
540 LINE (150,60)-STEP(44,-30),5
550 PAINT (155,59),5
560 S=(A*R^2)/2
570 PRESET(10,10): PRINT#1,"radio";R: PRESET(10,20): PRINT#1,"angulo ";A
580 GOSUB 590: RETURN
590 PRESET (10,120): PRINT#1," la superficie es ";S: RETURN
```



El código máquina: ese desconocido (cap. 3)

El mes pasado ya hicimos un pequeño programa que sumaba dos números. Esto nos sirvió para aprender algunos modos de direccionamiento: inmediato, directo e indirecto. Vimos también cómo podíamos usar el par de registros HL para este tipo de direccionamiento, de modo que el programa fuese independiente de los datos a tratar. Aunque sólo se ponía como ejemplo y para aprendizaje (resulta más fácil sumar en BASIC, sin duda alguna), si quisiésemos utilizarlo para alguna aplicación nos resultaría inservible, dado que sólo sabe sumar hasta 255 (debido a que usamos un *byte*. Vamos a ver como solucionar este problema haciendo que el número tenga toda la longitud que queramos.

La bandera de acarreo ('C').

Aunque ya dimos unas breves nociones del uso del *bit* de acarreo, vamos a explicar ahora la más importante en detalle, las otras que tienen ya las veremos más adelante, debido a que es necesaria para nuestro proceso.

Recordarán que cuando explicamos las operaciones en hexadecimal y lo que era un *byte*, vimos que teníamos 8 *bits* para operar; pero resulta que en determinados casos el resultado de una suma puede exceder esta capacidad. Si por ejemplo, sumanos '10110010b' y '11011010b', el resultado, como se puede ver en la figura 1, es de nueve *bits* excediendo los ocho de que dispone el acumulador.

Este *bit* extra no se desperdicia, sino que se almacena en la bandera de acarreo, de modo que si el resultado de la suma da un uno en la posición novena, esta bandera vale uno y si, por el contrario, la suma da cero, entonces valdrá cero. Esto por sí solo no tendría ninguna utilidad, si no fuese porque tenemos una instrucción especial que nos permite operar teniendo en cuenta esta bandera. La instrucción se llama ADC,

abreviatura de *'ADd with Carry'* (sumar con acarreo) y realiza la misma suma que vimos en el capítulo anterior; pero, además de dichas operaciones, suma el *bit* de acarreo al inferior del acumulador. Por tanto, si en dicho *bit* había un cero, la suma se realiza como ya hemos visto; pero si hay un uno, se incrementa el resultado en dicha cantidad, como se puede ver en la figura 2.

Esto hace que dispongamos del equivalente del 'y me llevo uno', tan usual en nuestros cálculos. Con esta operación disponemos de un método fácil de sumar números superiores a 8 *bit*. Como ejemplo, en la figura 3 vemos cómo se pueden realizar la suma de '1011001010011101b' y '0010110110011101b', ambos de 16 *bits*. En primer lugar se dividen ambos en dos grupos de 8 *bits*. A continuación, se suman las partes inferiores sin acarreo, ya que no nos llevamos nada de antes, a lo que es lo mismo, con cero de acarreo, esto es interesante tenerlo en cuenta ya que nos facilitará el trabajo posteriormente. De la suma de éstos se obtiene un *bit* de acarreo que se añade a la siguiente suma de las dos partes superiores, con lo que se obtiene todo el resultado. Evidentemente, es posible que en esta suma se vuelva a obtener un *bit* de acarrero. La precisión del cálculo y, por tanto, la posibilidad que se produzca error la determinamos nosotros indicando la cantidad de *bytes* a sumar.

Suma de dieciseis bits.

Para realizar un programa que sume en dieciseis *bits* vamos a aprender otro sistema de direccionamiento; el direccionamiento indexado. En este sistema se utiliza el registro IX o IV, como usamos el HL en el programa anterior; pero, además, se le puede sumar un número a esta dirección para hallar la que queremos usar. De este modo podemos hallar direcciones del tipo 'la tercera a partir de esta', o 'la que hay dos antes de aquella'. Siendo ésta y aquella el contenido del registro y el dos o el tres el número que se suma. Este sistema de direccionamiento es muy útil para direccionar matrices y en otros casos como el que estamos tratando. La instrucción que carga un número en el acumulador usando este sistema es la usual LD con el formato: 'LD A,(IX + desplazamiento)' o 'LD A,(IX + desplazamiento)' según estemos usando el registro IX o IV. También existen cuatro instrucciones específicas (dos con acarreo y dos sin acarreo) para realizar la suma con este direccionamiento. Son 'ADD A,(IX + des.)', 'ADD A,(IV + des.)', 'ADC A,(IX + des)' y 'ADC A,(IV + des.)'. En la figura 4 se da una tabla con los códigos decimales y hexadecimales de todas estas instrucciones. Un detalle importante es que todas ellas empiezan por 'DDh' (si se utiliza el registro IX) o 'FDh' (si se utiliza el registro IV). También hay que señalar que detrás de todas ellas debe haber un *byte* que indique el desplazamiento en el formato de complemento a dos que veremos posteriormente, de momento basta saber que este formato utiliza los números con signo, no sin él, como hemos tratado hasta ahora y considera positivos a los que tienen el *bit* más alto a cero, por lo que de momento podemos usar los mismo razonamientos empleados hasta ahora.

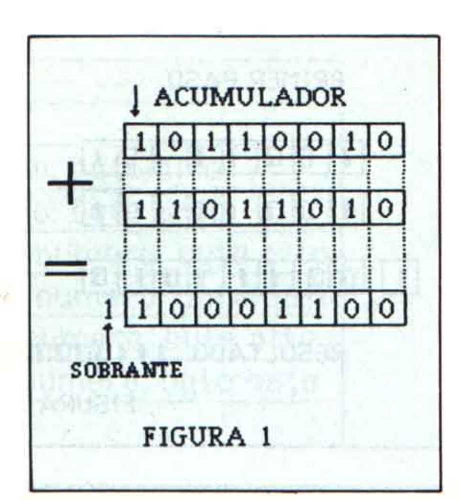


FIGURA 1

Por último, antes de realizar el programa se determinan las zonas donde se van a guardar los datos. En nuestro cso las asignaremos según el esquema indicado en la figura 5. El primer número a sumar se guardará en los *bytes* 40500 y 40501, con la parte inferior primero (luego veremos como meter el número desde BASIC). El segundo se almacena en 40502 y 40503. Por último, para meter el resultado se reservan el 40504 y 40505. Todos ellos con el mismo formato de la parte inferior primero.

Una vez determinado todo esto empezamos a desarrollar el programa. En primer lugar hay que determinar en el registro IX el valor que vamos a usar como base, 40500, que se hace con la instrucción 'LD IX, dato de 16 bit'. Recordando que el dato debe ir primero la parte baja y luego la alta. Una vez hecho esto se carga la parte inferior del primer dato usando una instrucción 'LD' pero con el registro índice más el desplazamiento. Este valdrá en este caso 0, ya que queremos leer la misma dirección 40500. A continuación, se le

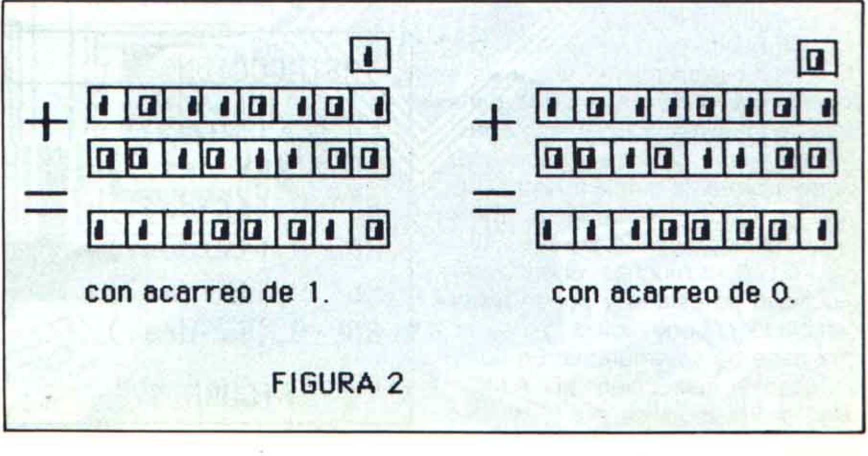


FIGURA 2

suma la otra parte baja, en esta ocasión sin acarreo, la dirección de esta es 40502, por lo que hay que sumar un desplazamiento de 2 al índice. Por último el resultado se guarda con el mismo sistema de direccionamiento pero usando la instrucción 'LD IX + des.), A' que realiza el procedimiento contrario a las anteriores, es decir, almacenar el contenido del acumulador en la dirección indicada por IX más el desplazamiento, esta instrucción es en hexadecimal 'DD 77 xx', donde xx es el desplazamiento. Para hacer la misma operación

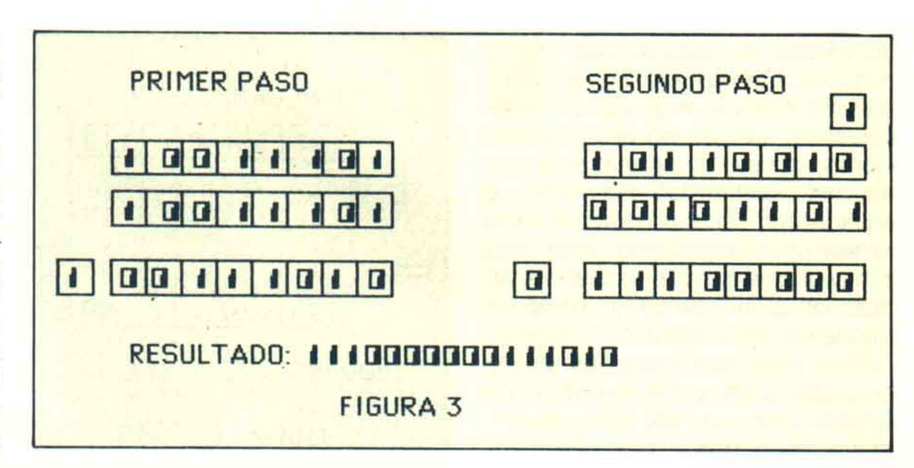
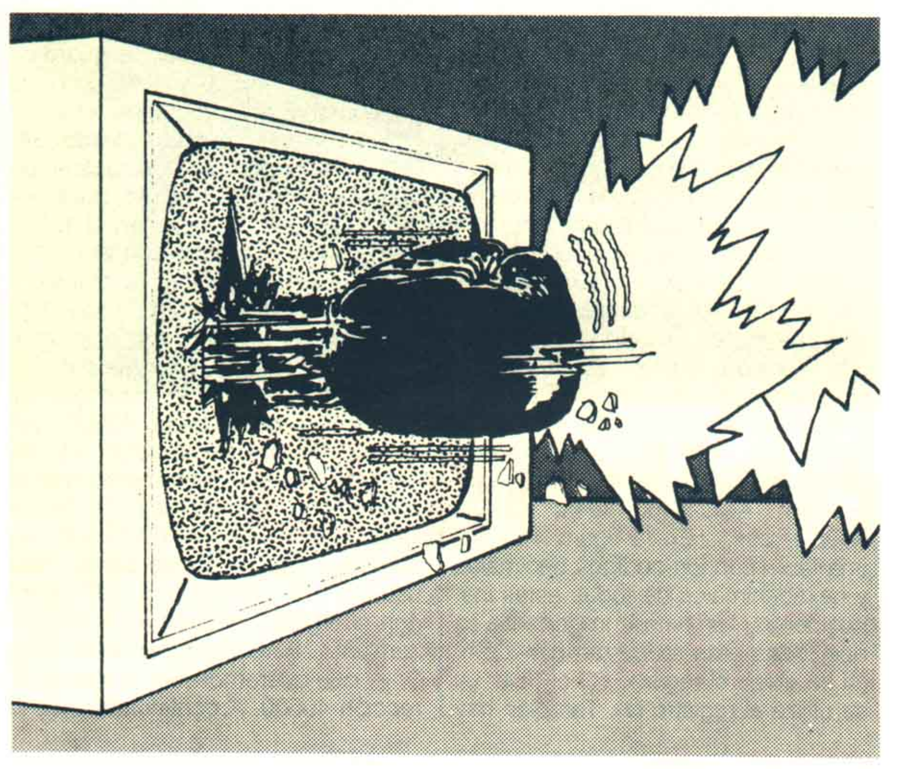


FIGURA 3



hay problemas. Esta hace que se carge la parte alta del primer dato (con un desplazamiento de 1) y seguidamente le sumamos el segundo dato pero esta vez con acarreo, es decir no se hace 'ADD' sino 'ADC' debido a que hay que tener en cuenta el acarreo. El resultado se almacena como la vez anterior, pero aumentando el desplazamiento en 1 y, al estar terminado el cálculo hacemos el RTS para volver al BASIC.

En la figura 6 se da el listado completo y sus correspondientes códigos hexadecimales y decimales. Todos los números de 16 *bits* han sido descompuestos en dos de 8 con el procedimiento de dividirlos entre 256. El resto es la parte inferior y el resultado (entero) es la parte alta. Por ejemplo, 43200 al dividirlo entre 256 da 168 de resultado (la parte superior) y 192 de resto, que es la parte inferior. Este cálculo se realiza así del mismo modo que en hexadecimal dividíamos entre 16 para hallar el se-

con IV basta cambiar el 'DD' por 'FD'. El despiazamiento ahora es 4 para formar 40504 que es la parte baja donde almacenar el resultado.

De resultas de esta operación el *bit* de acarreo se queda puesto o bajado en función de la suma. ¡Debe tener cuidado cuando lo vaya a utilizar!, ya que muchas operaciones modifican su estado y puede haber cambiado cuando usted lo use en otra parte de su programa. En nuestro caso la instrucción 'LD A,(IX + des.)' no la modifica por lo que no

INSTRUCCION	HEXADECIMAL	DECIMAL
LD A,(IX+des.)	DD 7E xx	221 126 xx
LD A,(IY+des.)	FD 7E xx	253 126 xx
ADD A,(IX+des.)	DD 86 xx	221 134 xx
ADD A,(IY+des.)	FD 86 xx	253 134 xx
ADC A,(IX+des.)	DD 8E xx	221 142 xx
ADC A,(IY+des.)	FD 8E xx	253 142 xx

FIGURA 4

40505 —	resultado. byte alto
40504 —	resultado. byte bajo
40503 —	segundo numero. byte alto
40502 —	segundo numero. byte bajo
40501 —	primer numero. byte alto
40500 —	primer numero. byte bajo

FIGURA 5

gundo decimal, en cierto modo es como si operásemos en base 256 (ya que 255 es el máximo número que cabe en un *byte*.)

Para introducir el programa en memoria usamos el cargador hexadecimal que ya describimos el mes pasado. En esta ocasión basta cambiar los DATA añadiendo al final el 256 que indica fin de datos. El listado modificado del cargador en BASIC se da en la figura 7.

Después de introducir la subrutina en lenguaje máquina, hay que meter los datos. Para ello, y teniendo en cuenta que van a ocupar 16 *bits*, es decir el valor máximo es 65535, se realiza el mismo proceso descrito antes de dividir entre 256. Si queremos sumar 13215 y 31100, haremos lo siguiente: 13215/256 es 51, es decir el valor alto a introducir en 40501 y el resto es 159 que va a 40500. Con 31100 se realiza el mismo proceso dando 121 y 124, que se almacenan respectivamente en 405003 y 40502. Todas las introducciones en menoria se hacen con la instrucción 'POKE dirección, dato', donde dirección es respectivamente 40500, 40501, etc. y dato son los números de 8 *bits* que hemos hallado como se ve en la figura 8.

Por fin se realiza la llamada a la rutina en lenguaje máquina (colocada en este caso en la dirección 40000) por medio de las instrucciones.

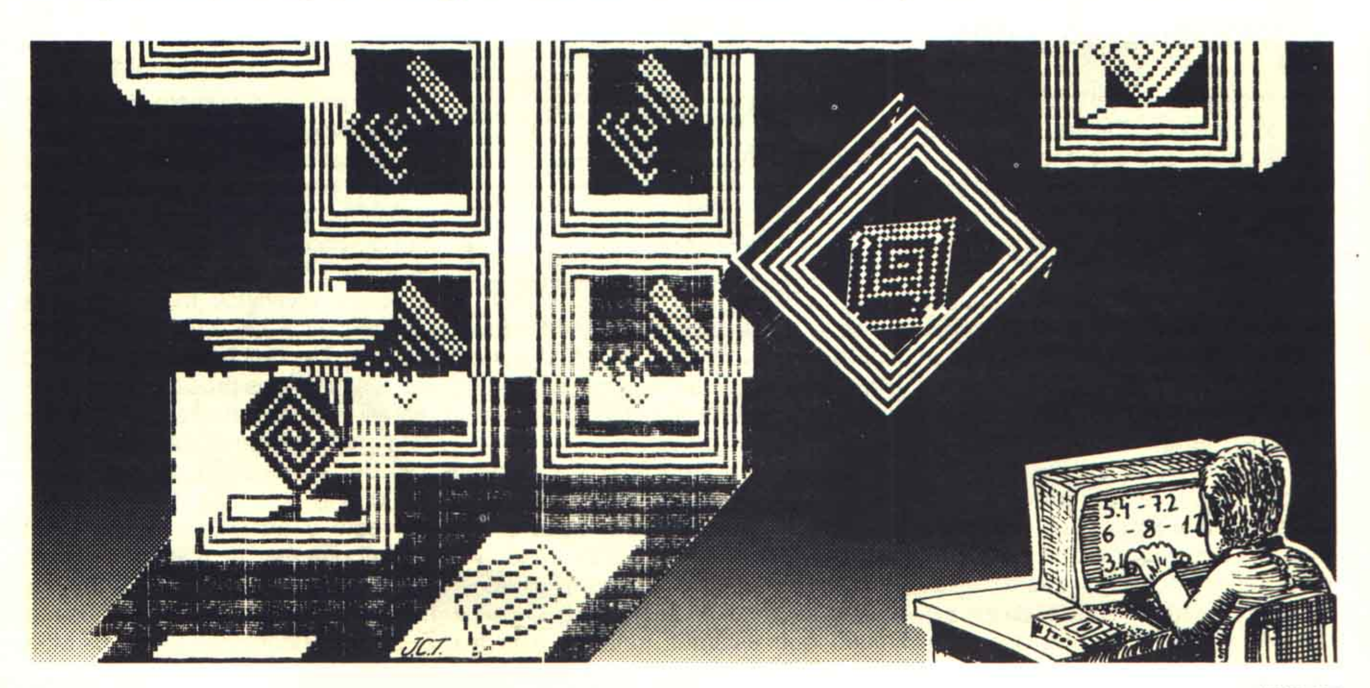
```
DEFUSRO=40000
PRINT USRO (1)
```

que ya explicamos el mes anterior.

Una vez hecho esto recuperamos la suma, que está compuesta en dos *bytes*; el 40504 y el 40505. Para hallar el valor total hacemos el proceso inverso del anterior, es decir, sacamos el valor de 40504 y le sumamos el de 40505 multiplicado por 256. Todo esto se puede hacer en una sola línea como sigue:

```
PRINT PEEK(40504) +
PEEK(40505)*256
```

Si hemos introducido los datos anteriormente comentados, obtendremos el resultado 44315, como era de esperar.



LISTADO		HEXADECIMAL	DECIMAL
LD	IX,40500	DD 21 34 9E	221 33 52 158
LD	A,(IX+0)	DD 7E 00	221 126 0
ADD	A,(IX+2)	DD 86 02	221 134 2
LD	(IX+4),A	DD 77 04	221 119 4
LD	A,(IX+1)	DD 7E 01	221 126 1
ADC	A(IX+3)	DD 8E 03	221 142 3
LD	(IX+5),A	DD 77 05	221 119 5
RET		C9	201

FIGURA 6

```
10 INPUT "DIRECCION INICIAL:";D
20 READ A
30 IF A=256 THEN PRINT "TERMINE":END
40 POKE D,A
50 D=D+1
60 GOTO 20
1000 DATA 221,33,52,158,221,126,0,221,134,2,221,119,4,221,126,1,221
,142,3,221,119,5,201,256
```

FIGURA 7

Números negativos: complemento a dos

Antes de continuar explicando las instrucciones de nuestro procesador es necesario que veamos el sistema utilizado para expresar números negativos en el Z-80, el complemento a dos, ya que nos va a hacer falta enseguida.

Hasta ahora hemos usado todos los *bytes* como si no tuviesen signo, por lo que todos los *bits* se utilizaban para expresar valores. Pero existe gran cantidad de casos en la que es necesario tener números negativos. Uno de ellos es en la indexación que acabamos de ver, en la que el desplazamiento si es positivo indica una memoria superior al indice, pero si es negativo entonces indica una memoria anterior, otro caso es el de los saltos relativos al programa que veremos a continuación, que son hacia delante o hacia atrás según el signo del número que se use. Por estas razones hace falta saber como expresar datos con signo.

```
POKE 40500,159
POKE 40501,51
POKE 40502,124
POKE 40503,121
```

FIGURA 8

Cuando queremos expresar un número con signo, es necesario que podamos tener un sistema de diferenciar rapidamenta entre uno y otros. Este puede ser (y es) el estado de un *bit*. Si el *bit* superior de la cantidad que estamos tratando está a cero, podemos decir que ésta es positiva y si esta a uno, podemos decir que es negativa.

Si el resto del número lo dejamos como siempe, obtendremos el formato denominado magnitud más signo, en éste la única diferencia entre un dato positivo y uno negativo está en el *bit* más superior que estará a cero o a uno respectivamente. Según esto y si operamos con números de un *byte*, '00010110b' vale 22 y '10010110b' es -22. La cantidad de esto se complica algo si los números están en hexadecimal ya que entonces el primero vale '16h' y el segundo '96h'. Como ayuda hay que saber que, para cualquier sistema de expresión de números son signo, si se expresa en hexadecimal y es igual o superior a '80h', entonces en negativo.

Pero este sistema tiene algunos fallos el primero de ellos, y quizá el más avidente, se puede ver inmediatamente: '00000000b' es cero, como ya hemos visto, pero resulta que '10000000b' tambien es cero, dado que lo que se nos indica es que es -0, pero esto, según aprendimos en el colegio es lo mismo que cero ya que no tiene signo. Otro problema no tan evidente radica en el hecho de que los ordenadores no les resulta fácil trabajar con este sistema, por lo que hay que recurrir a otro. Este otro es el denominado complemento a 1.

Para pasar un número de positivo a negativo en el caso anterior bastaba con cambiar la posición del *bit* más superior, otro planteamiento puede ser el el de cambiar todos los *bits* del número. Por tanto para pasar el número '00000100b' (4) a negativo basta invertirlos todos y queda que menos 4 es '11111011b'. Este sistema, denominado complemento a 1, simplifica los procesos que debe realizar el microprocesador para sumar y restar, pero sigue existiendo el fallo de la existencia de dos ceros: '0000000b' y '1111111b'. Para arreglar esto y además simplificar aún más el proceso al ordenador, surgió el sistema denominado de complemento a 2. Este es el que usa el Z-80 que incorporan todos los MSX

y al que prestamos más atención.

En este caso, el proceso que se realiza para pasar un número de positivo a negativo consiste en invertir todos los *bits* como en el complemento a 1 y luego se le suma uno. Por tanto, el -4 se halla invirtiendo '00000100b', que e la expresión buscada. Si intentamos hallar -0, veremos que la inversa de '00000000b' es '11111111b', que al sumarle 1 da '00000000b', es decir, 0 es igual a -0. En este último proceso puede resultar (como en el caso del cero) que se ponga el *bit* de acarreo, pero en este caso específico no tiene ninguna utilidad y, por tanto, se tiene que despreciar.

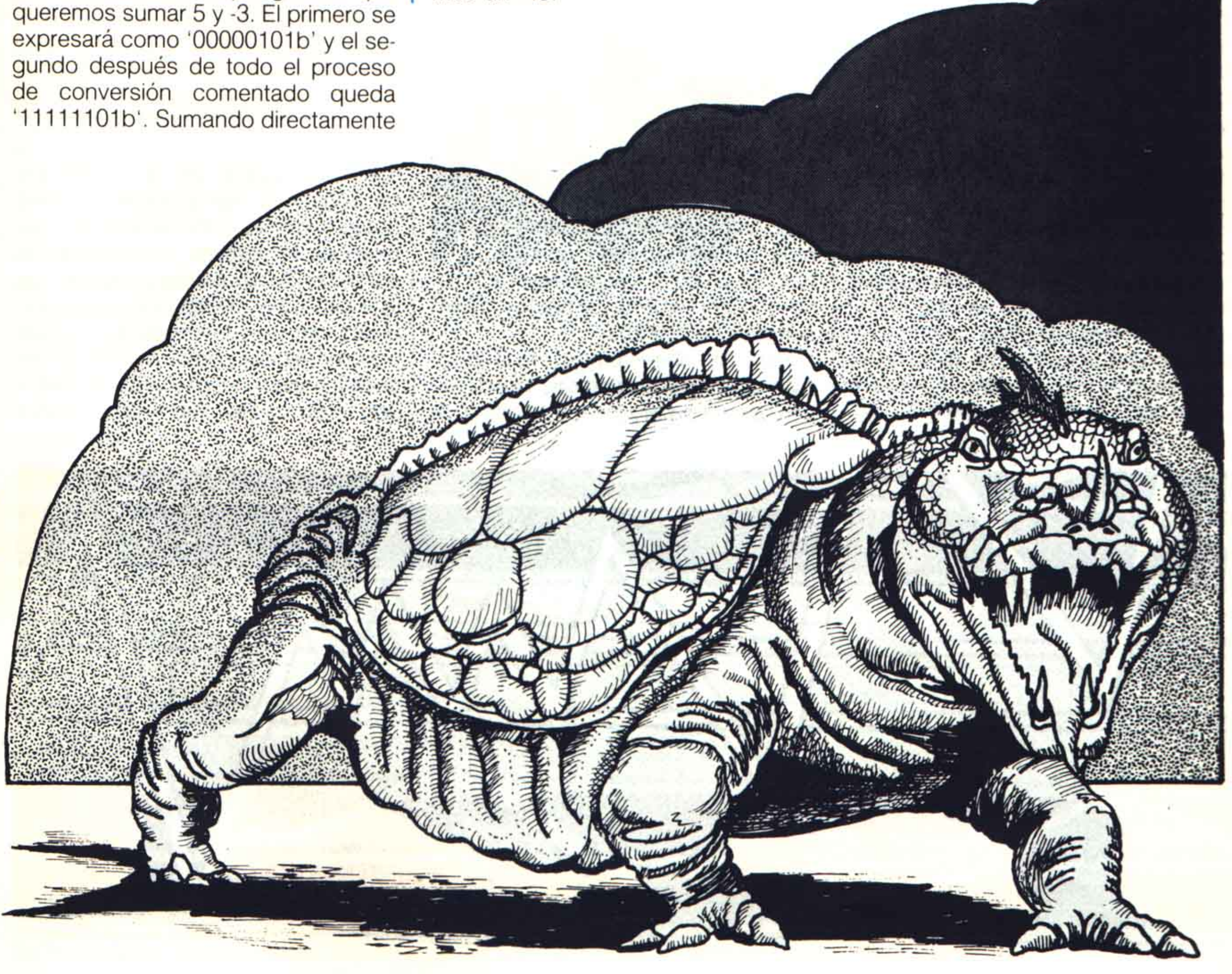
Otra ventaja del complemento a dos reside en la rapidez y sencillez de los cálculos. Supongamos que queremos sumar 5 y -3. El primero se expresará como '00000101b' y el segundo después de todo el proceso de conversión comentado queda '11111101b'. Sumando directamente queda: '00000010b' que es, sorprendentemente, 2 (en este caso también se pone el *bit* de acarreo pero, como antes, hay que despreciarlo). Con lo que queda claro que las operaciones matemáticas se pueden realizar como si fuesen sin signo.

Por último, para pasar un número negativo a positivo en este formato hay que realizar el proceso inverso al de pasar de positivos a negativos pero respetando el orden, es decir, primero se resta uno al número y luego se invierte, si cambiásemos el orden de las operaciones, nos daría el resultado erróneo. Así pues '11010010b' para pasarlo a positivo se le resta 1, que da '11010001b' y luego se invierte obteniendo '00101101b', lo que nos permite averiguar que teníamos un -45.

Ampliando la suma

Una vez visto como operar en 8 y 16 *bits*, vamos a desarrollar un sistema que nos permita operar con la precisión que queramos de un modo sencillo y sin tener que añadir un nuevo bloque de instrucciones por cada *byte* que sumemos. Para poder realizarlo hemos de acudir a una estructura que ya hemos visto y usado ampliamente en BASIC: los bucles.

Está claro que si podemos hacer un ciclo dentro del cual se sume un *byte*, será suficiente repetirlo tantas veces como posiciones de memoria ocupen los datos. En nuestro caso



este proceso se hará tres veces, pero su modificación o cualquier otro número es inmediata.

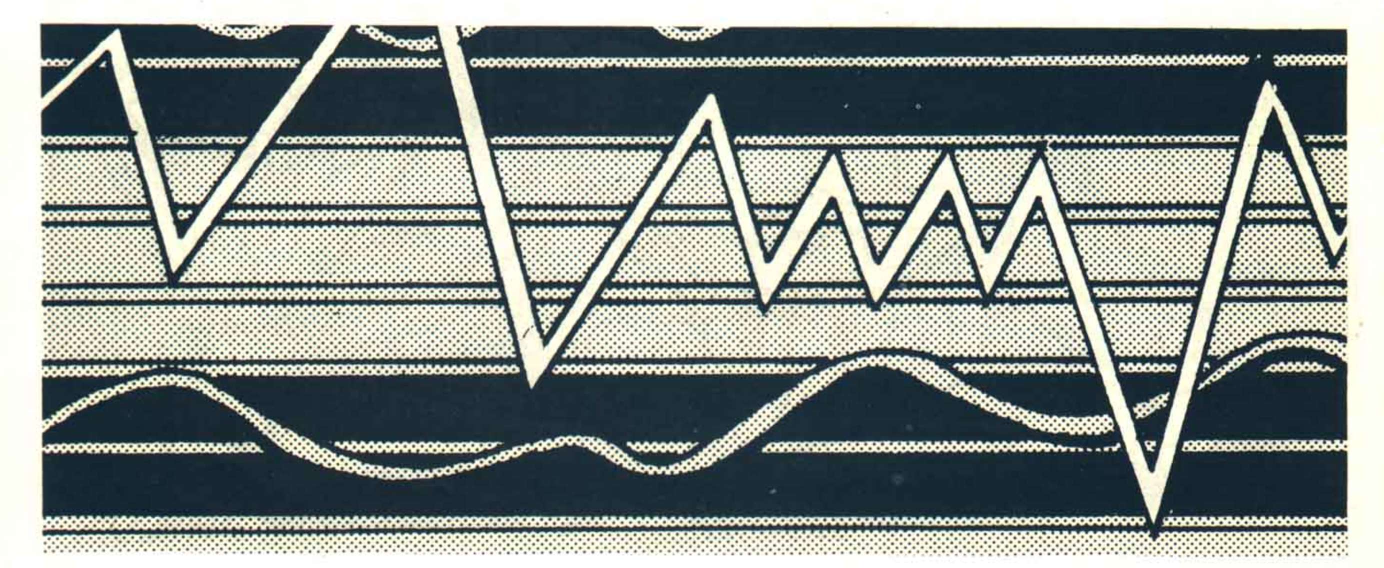
Este proceso de suma dentro del bucle puede ser el que ya hemos usado en la de 16 *bits*, en la que se utiliza el registro de índice IX para cargar el *byte* correspondiente del primero número y luego se le sumaba el del segundo con acarreo. Esto implica que la primera adicción, que en el caso anterior la hacíamos sin acarreo, ahora se hace con él,por lo que deberemos tener cuidado en ponerle a cero antes de entrar en el bucle para que no influya. Esto se hace con dos instrucciones: 'SCF' ('37h', 55 decimal) que pone la bandera de acarreo y 'CCF' ('3Fh, 63 decimal) que se invierte su estado actual poniéndola a cero, ya que la acabamos de poner a uno. Si estuviese previamente a uno, la pondría a cero. La razón de que utilizemos dos instrucciones para una cosa tan sencilla se debe a que el Z–80 no tiene una instrucción específica para ponerla a cero, pero sí para ponerla a uno e invertirla como hemos hecho.

Otra cosa que sufre una modificación es la forma en que usábamos el registro índice. Antes, éste se quedaba fijo y lo único que variaba era el desplazamiento que se le sumaba, esto no puede ser hecho así en este caso porque toda la suma se realiza dentro del bucle y, por tanto, el desplazamiento siempre será igual. Como el ordenador no nos permite que este desplazamiento lo tengamos almacenado en una memoria que se pueda modificar, sino que tiene que ser inmediato, hemos de recurrir a otro sistema.

40508 →	resultado. byte alto
40507 →	resultado. byte medio
40506 →	resultado. byte bajo
40505 →	segundo numero. byte alto
40504 →	segundo numero. byte medio
40503 →	segundo numero. byte bajo
40502 →	primer numero. byte alto
40501 →	primer numero. byte medio
40500 →	primer numero. byte bajo

FIGURA 9

Para ello observemos una disposición válida de la memoria donde se van almacenando los números, en la figura 9 se puede ver que cada uno de ellos está almacenado en tres *bytes* y que la parte inferior del número es la primera y la superior la última. En base a esto podemos ver que los *bytes* del segundo sumando están tres por detrás de los correspondientes del primero y los del resultado esán desplazados 6. Siempre, los *bytes* que tiene estas



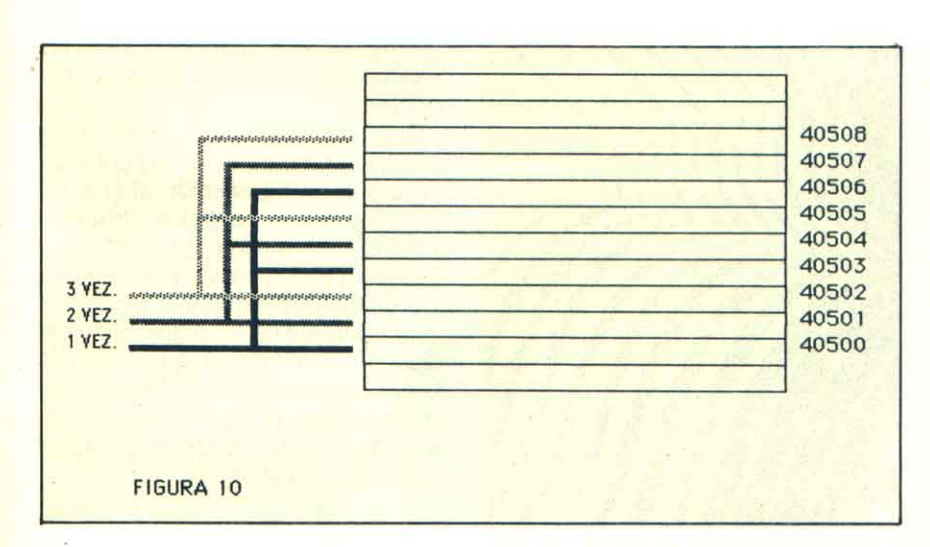
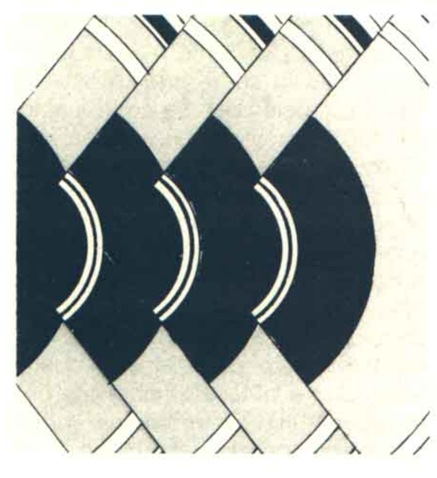


FIGURA 10

diferencias son los del mismo peso (el primero, el segundo o el tercero) y podemos deducir que si sabemos cual de los tres tenemos que sumar y tenemos la dirección correspondiente a ese en el primer número, para hallar la el segundo basta sumarle tres a la anterior y para el resultado se le suma seis.

Esto nos permite simplificar el proceso, ya que si hacemos que el registro índice apunte siempre al *byte* correspodiente del primer número con un desplazamiento de cero, los otros se hallan modificando este desplazamiento pero que valga 3 y 6 como se ve en la figura 10.

Para poder realizar esto, hay que incrementar IX después de cada ejecución del bucle de modo que apunte al siguiente *bytes*. Para ello usamos una instrucción que se hace exactamente esto, se llama 'INC IX' y permite aumentar en uno el valor de este registro.



A continuación hay que desarrollar un sistema que nos permita salir del contador cuando termine la suma. Para ello sería necesario que tuviésemos un contador que nos diga cuantas veces hemos hecho el bucle y cuantas nos quedan por hacerlo. En un principio puede parecer que valdría el IX que ya estamos utilizando, pero existen problemas para poder comprobar su valor y, además, dada la disposición interna del microprocesador, resulta mucho más fácil hacer un contador que se decremente hasta llegar a cero, en cuyo caso hay que mirar esto. Un caso que puede ser usado es el de tener este contador en memoria y modificarlo cada vez, pero eso nos obligaría a cargarlo, restarle uno, meterle en memoria una vez modificado y comprobar si vale cero. La mayor parte de estas operaciones puede ser evitada si almacenamos este contador en un registro. Pero como el acumulador 'A' ya está ocupado recurriremos a otro. Por ejemplo el 'B'.

Del mismo modo que en el BASIC se les asigna un valor inicial a los bucles, aquí también hay que hacerlo, para ello cargaremos un 3 en este registro por medio de la instrucción 'LD B, número', estando el número en modo inmediato. El código de esta operación es 6, o si lo prefieren en hexadecimal '06h' y debe ir seguido el valor.

Por tanto ya tenemos el sistema de sumar los dos números incrementando el registro índice y como asignar el valor inicial al contador. Ahora falta saber como decrementar este y comprobar si ha llegado a cero para salir del bucle 'DEC B' (5 decimal, '05h' hexadecimal) que automáticamente resta 1 a este registro y pone la bandera 'Z' (cero) a uno si vale cero.

Por último hay que comprobar esta bandera y si no vale cero tiene que volver al principio, (donde se

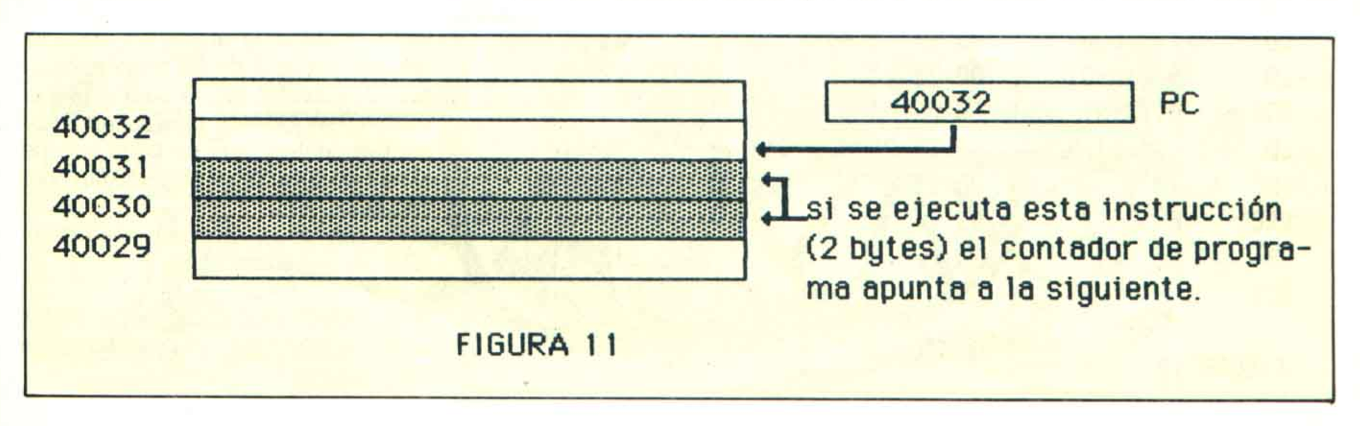


FIGURA 11



carga el *byte* correspondiente del primer sumando). Para ello usamos otra nueva instrucción: 'JR NZ, des' ('20h' ó 32 decimal), que significa *"Jump Relative if Not Zero"* (salto relativo si no es cero). Su modo de funcionamiento es el que sigue. Si la bandera 'Z' está a cero, sigue con la siguiente instrucción pero sino, le suma el desplazamiento, en complemento a dos, al contador del programa (PC) y salta a la dirección que indique el resultado. Aquí hay que tener mucho cuidado con una cosa que crea muchos problemas y errores. El contador de programa siempre apunta al *byte* siguiente al que se ha cargado por última vez. De modo que si, como se ve en la figura 11, esta instrucción estaba en la memoria 40030, y dado que ocupa dos *bytes*, la siguiente empezará en la 40032 y el contador de programa tendrá este valor cuando ejecutemos nuestra instrucción de compración y salto. Es decir, el 'PC' **siempre** apunta al principio de la siguiente instrucción. Por tanto, si en el caso que hemos puesto por ejemplo, queremos saltar a la 40020, habrá que saltar 40032–40020, lo que da -12 ya que el salto es hacia atras y por tanto negativo. Esto nos obliga a ir contando los *bytes* que ocupa cada instrucción y cual es su dirección, pero no preocupe, cosas más difíciles se han hecho.

Con esto ya tenemos hecha nuestra rutina como se ve en la figura 12. En la que se ha añadido al final un RTS para que el control se devuelva al BASIC. Al terminar de ejecutarse, el registro B contendrá 0, ya que era la condición de salida y el registro IX apuntará a 40503 (el primer byte del segundo número) ya que después de ejecutarse el bucle por última vez se incrementa de nuevo. Un equivalente de esto es hacer en BASIC el siguiente programa:

```
5  B = 3
10 X = 40500
20 instrucciones de suma dentro del bucle
30 X = X + 1
40 B = B -1
40 B = B -1
50 IF B  0 THEN GOTO 20
60 END
```

Que si le ejecuta (poniendo en Rem en la instrucción 20) verá que deja como valores finales los que hemos dicho.

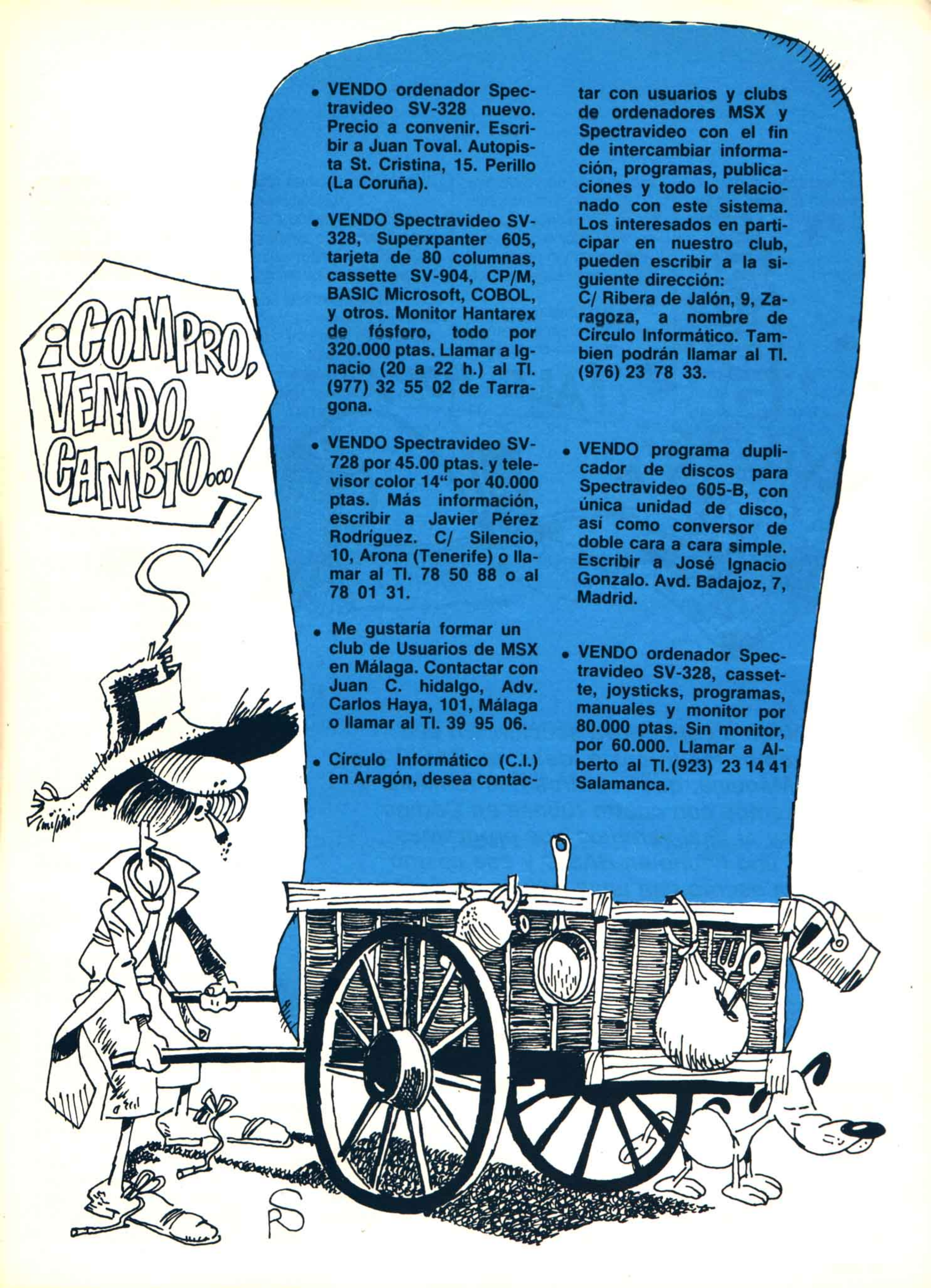
El sistema de llamada y de introducción de datos son igual que en el proceso anterior, pero esta vez los números son de tres *bytes*, con lo que se nos permite valores mayores pero también hay que descomponerlos y componerlos en tres. Para ello al igual que antes dividiamos entre 256, ahora tenemos que hacer entre 65536. El resultado es el *byte* de mayor valor y con el resto (que debe ocupar 2 bytes como antes) debemos realizar el mismo proceso de dividir entre 256 y tomar el resultado y el resto. Fíjese que ahora las direcciones de empiece de los números son 40500, 40503 y 40506 para el primer dato, el segundo y el resultado respectivamente y que en cada uno de ellos hay que hacer tres POKE consecutivos.

Por último, el proceso de obtención del resultado es:

PRINT PEEK (40506) + PEEK (40507)*256 + PEEK(40508)* 65536

LISTADO		HEXADECIMAL	DECIMAL
SCF		37	55
CCF		3F	63
LD	B,3	06 03	6 3
LD	IX,40500	DD 21 34 9E	221 33 52 158
LD	A,(IX+0)	DD 7E 00	221 126 0
ADC	A(IX+3)	DD 8E 03	221 142 3
LD	(IX+6),A	DD 77 06	221 119 6
INC	IX	DD 23	221 35
DEC	B	05	5
JR	NZ,-14	20 F2	32 242
RET		C9	201

FIGURA 12



- VENDO ordenador Spectravideo SV-328 nuevo. Precio a convenir. Escribir a Juan Toval. Autopista St. Cristina, 15. Perillo (La Coruña).

- VENDO Spectravideo SV-328, Superxpanter 605, tarjeta de 80 columnas, cassette SV-904, CP/M, BASIC Microsoft, COBOL, y otros. Monitor Hantarex de fósforo, todo por 320.000 ptas. Llamar a Ignacio (20 a 22 h.) al Tl. (977) 32 55 02 de Tarragona.

- VENDO Spectravideo SV-728 por 45.00 ptas. y televisor color 14" por 40.000 ptas. Más información, escribir a Javier Pérez Rodríguez. C/ Silencio, 10, Arona (Tenerife) o llamar al Tl. 78 50 88 o al 78 01 31.

- Me gustaría formar un club de Usuarios de MSX en Málaga. Contactar con Juan C. hidalgo, Adv. Carlos Haya, 101, Málaga o llamar al Tl. 39 95 06.

- Círculo Informático (C.I.) en Aragón, desea contactar con usuarios y clubs de ordenadores MSX y Spectravideo con el fin de intercambiar información, programas, publicaciones y todo lo relacionado con este sistema. Los interesados en participar en nuestro club, pueden escribir a la siguiente dirección: C/ Ribera de Jalón, 9, Zaragoza, a nombre de Círculo Informático. Tambien podrán llamar al Tl. (976) 23 78 33.

- VENDO programa duplicador de discos para Spectravideo 605-B, con única unidad de disco, así como conversor de doble cara a cara simple. Escribir a José Ignacio Gonzalo. Avd. Badajoz, 7, Madrid.

- VENDO ordenador Spectravideo SV-328, cassette, joysticks, programas, manuales y monitor por 80.000 ptas. Sin monitor, por 60.000. Llamar a Alberto al Tl. (923) 23 14 41 Salamanca.



Bienvenidos a esta nueva sección. En ella publicaremos los trucos e ideas, en BASIC o en Código Máquina, que vosotros nos remitais. Y empezaremos con cuatro rutinas en Código Máquina y al final veremos dos programas ejemplo, uno hecho en BASIC y ese mismo programa escrito con unas rutinas en Código Máquina.

Profundizando en la memoria ROM (Read Only Memory *–Memoria de sólo lectura) existe un complejo programa compuesto por gran cantidad de subrutinas, que son utilizadas por el intérprete BASIC. Estas solamente pueden ser aprovechadas por el usuario en los programas en Código Máquina puesto que la mayoría de ellas no son directamente accesibles desde el BASIC.*

Vamos a ver el uso de cuatro de estas subrutinas y como funcionan.

Las dos primeras son equivalente a los comandos del BASIC, VPOKE y VPEEK. La primera de ellas comienza en la posición de memoria 4D hexadecimal (o 77 decimal), su función es la de escribir un dato en una posición de memoria de video (VRAM).

Desde un programa en Código Máquina, la utilizaremos de la siguiente forma:

Primero, cargamos el acumulador con el dato que queremos escribir en la memoria VRAM. A continuación cargaremos en el registro HL, la dirección de la memoria de video en la que vamos a escribir ese dato y por último, haremos una llamada a la rutina. Esto se hace con la instrucción CALL.

La segunda rutina a comentar, comienza en la dirección de memoria 4A hexadecimal (74 decimal). Su función es la de leer un dato de una de las posiciones de la memoria de video y se utiliza de forma similar a la rutina anterior.

Primero, cargamos en el registro HL la posición de memoria de la cual queremos leer el dato, luego hacemos la llamada a la rutina (CALL) y por último, la rutina almcena el dato en el acumulador.

A coontinuación, vamos a comentar dos rutinas que sirven para mover bloques de la memoria de video a la memoria principal y viceversa.

La primera de ellas, comienza en la dirección 59 hexadecimal (89 decimal) y permite mover un bloque de la memoria video a la memoria principal. La segunda, empieza en la dirección 5C hexadecimal (92 decimal) y realiza la función inversa a la anterior, es decir, mueve un bloque de la memoria principal a la memoria de video.

Estas dos rutinas se utilizan de idéntica forma. Primero, se carga el registro HL con la dirección de comienzo del bloque a mover, luego se carga el registro DE con la dirección a la que va

destinado dicho bloque. Después se carga el registro BC con la longitud del bloque a mover y por último llama a la rutina correspodiente. Estas rutinas tienen una aplicación muy útil, sobre todo a la hora de almacenar pantallas en la memoria y recuperarlas más tarde.

Con ellas podrá crear programas de laberintos, misiones de combate, etc, donde hacen falta más pantalla para darle más movilidad.

Para observar las grandes posibilidades de estas rutinas, veamos los siguientes programas.

Los dos realizan la misma función, pero una bastante más rápido que el otro. Esto se debe a que emplea las rutinas en Código Máquina (línea 180 del prograa 1).

El programa 2 que hace la misma función es bastante más lento debido a que está hecho integramente en BASIC.

Las dos rutinas desensambladas, que se utilizan en el programa 1 son las siguientes:

1. Movimiento del contenido de la VRAM a la memoria principal.

```
LD HL, 0
LD DE, 40000
LD BC 16384
CAL 89
RET
```

2. Movimiento del contenido de la memoria principal a la VRAM.

```
LD HL, 40000
LD DE, 0
LD BC, 16384
CALL 92
RET
```

Pero para que esto no caiga en saco roto, lo mejor es aplicarlas a ejemplos prácticos, por lo que aconsejamos hacer programas y utilizarlas.

PROGRAMA 1

```
10 FOR A=38900! TO 38925!:READ C:POKE A,C:NEXT:DEFUSR=38900!:DEFUSR1=38913!
20 SCREEN2:OPEN"grp:"AS#1
30 CIRCLE (128,91),80,15,,,1.3
40 PAINT STEP(0,0)
50 CIRCLE (128,91),60,10,,,1.3
60 PAINT STEP(0,0),10
70 CIRCLE (128,91),40,6,,,1.3
80 PAINT STEP(0,0),6
90 CIRCLE (128,91),20,2,,,1.3
100 PAINT STEP(0,0),2
110 PSET(1,180):PRINT#1,"PULSA UNA TECLA PARA ALMACENAR"
120 IF INKEY$=""THEN 120
130 A=USR(A)
140 SCREEN 0:LOCATE 1,20:PRINT"PULSA UNA TECLA PARA RECUPERAR"
150 IF INKEY$=""THEN 150
160 SCREEN 2:A=USR1(A)
170 GOTO 170
180 DATA 33,0,0,17,64,156,1,0,64,205,&h59,0,201,33,64,156,17,0,0,1,0,64,205,&h5c
,0,201
```

PROGRAMA 2

```
1 N=40000!
20 SCREEN2:OPEN"grp:"AS#1
30 CIRCLE (128,91),80,15,,,1.3
40 PAINT STEP(0,0)
50 CIRCLE (128,91),60,10,,,1.3
60 PAINT STEP(0,0),10
70 CIRCLE (128,91),40,6,,,1.3
80 PAINT STEP(0,0),6
90 CIRCLE (128,91),20,2,,,1.3
100 PAINT STEP(0,0),2
110 PSET(1,180):PRINT#1,"PULSA UNA TECLA PARA ALMACENAR"
120 IF INKEY$=""THEN 120
130 FOR A%=0 TO 16383:X%=VPEEK(A%):POKE N+A%,X%:NEXT
140 SCREEN 0:LOCATE 1,20:PRINT"PULSA UNA TECLA PARA RECUPERAR"
150 IF INKEY$=""THEN 150
160 SCREEN 2:FOR A%=0 TO 16383:X%=PEEK(N+A%):VPOKEA%,X%:NEXT
170 GOTO 170
```

Rincón del lector

COMMODORE MSX

He oído repetidos rumores de que Commodore va a comercializar un ordenador MSX. Tiene fundamento esta suposición, en caso afirmativo, me aconsejarían esperar antes de comprarme un ordenador en vista del gran éxito que ha tenido el C-64.

Por último quiero hacer unas sugerencias a esta joven revista. No caigan en la tentación que han caido otras revistas de llenar sus páginas de programas de juegos, por ahora no han caido en ella. Otra sugerencia es que podrían incluir en sus páginas un apartado para enseñar a programar a los principiantes tanto BASIC como código máquina y por último, podrían publicar esquemas para mejorar el ordenador como podrían ser un modem, etc.

José López Agulló

Lo que podemos asegurar en la cuestión del Commodore MSX es que tal aparato no existe ni existirá. Sería absurdo que los americanos, tan defensores de su mercado como lo son, permitan que se implante el estándar japonés en su territorio. Prueba de ello es la lucha constante que hay en aquel país en el apartado de los ordenadore domésticos. Esto les lleva a mejorar sus ordenadores para poder competir en precio y prestaciones frente a los MSX.

Lo que no harán nunca será pedir la licencia para fabricar ordenadores Commodore MSX.

Las observaciones que apuntan están muy bien y se agradecen. Nosotros procuraremos no llenar las páginas de programas, prueba de ello es que en números anteriores no hemos sobrepasado los 7 programas. Ahora bien, el buen tiempo y el verano, en muchos casos quitan las ganas de leer, por lo que nos veremos en la situación de introducir algún que otro programa más. Esto no ha de preocuparte, puesto que el esquema que hemos desarrollado a lo largo de los primeros números, lo vamos a optimizar y mejorar sensiblemente con la apertura de nuevas secciones.

En cuanto a las secciones educativas es un tema que estamos intentando desarrollar. Hemos iniciado una serie, como habrás podido comprobar, de código máquina. El motivo de iniciar con este tema y no con el BASIC, se basa en el respeto que imponen los contenidos desconocidos. Por ello, en cuanto se hayan publicado algunas partes más de esta interesante serie y cuando estemos seguro de que, por lo menos, la mayoría de los usuarios han perdido ese respeto o miedo que este tema impone, iniciaremos la serie de programación en BASIC, que la alternaremos con la primera, para de esta forma completar un poco los conocimientos de nuestros lectores.

En la última cuestión entraremos en futuros números, ya que actualmente es prematuro hablar sobre ello.

DIRECTOR:
Juan Arencibia
COORDINADOR EDITORIAL:
Emiliano Juárez
REDACCION:
Fernando García, Santiago Gala, Ricardo García, Teresa Aranda, Francisco Mancera.
DISEÑO:
Ricardo Segura y Benito Gil.
Editada por
PUBLINFORMATICA S.A.
PRESIDENTE:
Fernando Bolín.
DIRECTOR EDITORIAL:
Norberto Gallego.
Administración:
PUBLINFORMATICA
GERENTE DE CIRCULACION Y VENTAS:
Luis Carrero.
PRODUCCION:
Miguel Onieva.
DIRECTOR DE MARKETING:
Antonio Gonzalez.
SERVICIO AL CLIENTE:
Julia Gonzalez.
Tel. 733 79 69
ADMINISTRACION:
Miguel Atance,
JEFE DE PUBLICIDAD:
María José Martín.
DIRECCION Y REDACCION:
C/ Bravo Murillo, 377 - 5° A
Tel. 733 74 13
28020 MADRID
PUBLICIDAD Y ADMINISTRACCION:
C/ Bravo Murillo 377 - 3° E
Tel. 733 96 62-96
Publicidad en Madrid:
Fernando Hernando
Publicidad en Barcelona:
Olga Martorell
C/ Pelayo, 12
Tel. (93) 301 47 00 Ext. 27-28.
08001 BARCELONA
Deposito Legal: M. 16.755-1985
Impreso en Héroes S.A.
C/ Torrelara, 8. 28016-MADRID.
Distribuye:
S.G.E.L. Avda. Valdelaparra s/n. Alcobendas (Madrid).

SUSCRIPCIONES:
Rogamos dirija toda la correspondencia relacionada con suscripciones a:
MSX
EDISA: Tel. 415 97 12
C/López de Hoyos, 141-5.°
28002 MADRID
(Para todos los pagos reseñar solamente MSX)
Para la compra de ejemplares atrasados dirijanse a la propia editorial
MSX
C/Bravo Murillo, 377-5.° A
Tel. 733 74 13 28020 MADRID

Si deseas colaborar en MSX remite tus artículos o programas a Bravo Murillo 377, 5.° A. 28020 Madrid. Los programas deberán estar grabados en cassette y los artículos mecanografiados.
A efectos de remuneración, se analiza cada colaboración aisladamente, estudiando su complejidad y calidad.